Security Audit Compliance For Cloud Computing

by

Frank Hans-Ulrich Doelitzscher

A thesis submitted to Plymouth University in partial fulfilment for the degree of

DOCTOR OF PHILOSOPHY

School of Computing, Communication & Electronics

In collaboration with

University of Applied Sciences Furtwangen, Germany

February 2014

COPYRIGHT STATEMENT

Abstract

Security Audit Compliance For Cloud Computing

Frank Hans-Ulrich Doelitzscher
MSc

Cloud computing has grown largely over the past three years and is widely popular amongst today's IT landscape. In a comparative study between 250 IT decision makers of UK companies they said, that they already use cloud services for 61% of their systems. Cloud vendors promise "infinite scalability and resources" combined with on-demand access from everywhere. This lets cloud users quickly forget, that there is still a real IT infrastructure behind a cloud. Due to virtualization and multi-tenancy the complexity of these infrastructures is even increased compared to traditional data centers, while it is hidden from the user and outside of his control. This makes management of service provisioning, monitoring, backup, disaster recovery and especially security more complicated. Due to this, and a number of severe security incidents at commercial providers in recent years there is a growing lack of trust in cloud infrastructures.

This thesis presents research on cloud security challenges and how they can be addressed by cloud security audits. Security requirements of an Infrastructure as a Service cloud are identified and it is shown how they differ from traditional data centres. To address cloud specific security challenges, a new cloud audit criteria catalogue is developed. Subsequently, a novel cloud security audit system gets developed, which provides a flexible audit architecture for frequently changing cloud infrastructures. It is based on lightweight software agents, which monitor key events in a cloud and trigger specific targeted security audits on demand - on a customer and a cloud provider perspective.

To enable these concurrent cloud audits, a Cloud Audit Policy Language is developed and integrated into the audit architecture. Furthermore, to address advanced cloud specific security challenges, an anomaly detection system based on machine learning technology is developed. By creating cloud usage profiles, a continuous evaluation of events - customer specific as well as customer overspanning - helps to detect anomalies within an IaaS cloud. The feasibility of the research is presented as a prototype and its functionality is presented in three demonstrations. Results prove, that the developed cloud audit architecture is able to mitigate cloud specific security challenges.

Contents

List of Figures VII

List of Tables XI

Acknowledgements XV

Authors Declaration XVII

1 Introduction & Overview **1**

1.1 Introduction 2

1.2 Aims & Objectives 3

1.3 Thesis Structure 6

1.4 Security Audit as a Service Research Grant 9

2 Review of Cloud Computing **11**

2.1 Introduction 12

2.2 Evolution of Cloud Computing 12

2.3 Definition of Cloud Computing 15

2.4 Cloud Deployment & Service Models 17

2.5 Cloud Computing Architecture 21

2.6 Roles in Cloud Computing 23

2.7 Related Technologies to Cloud Computing 26

2.8 Commercial Cloud Landscape 27
2.9 Advantages and Disadvantages of Cloud Computing 30
2.10 Summary 33

3 Cloud Infrastructure Audit 35

3.1 Introduction 36
3.2 Security Concerns in Cloud Computing 37
3.3 Related Work 40
3.4 Cloud Computing Security Issues 46
3.4.1 Cloud Computing vs. Classic IT Outsourcing 47
3.4.2 Amplified Cloud Security Problems 49
3.4.3 Specific Cloud Security Problems 53
3.5 Security Audits in Clouds 60
3.5.1 IT Security Audit Types 61
3.5.2 Classic IT Security Audits vs. Cloud Audits 62
3.5.3 Cloud Audit Use Cases 69
3.5.4 Cloud Audit Test Criteria Catalogue 80
3.6 Summary 85

4 Security Audit as a Service (SAaaS) 89

4.1 Introduction 90
4.2 Related Work 91
4.3 SAaaS Use Cases 95
4.4 Cloud Audits Using Agents 97
4.5 SAaaS Event Processing 98
4.6 SAaaS Agents 100
4.7 Evaluation: SAaaS Architecture 109
4.8 Summary 116

5 A Cloud Audit Policy Language **117**

5.1 Introduction . . . 118

5.2 Related Work . . . 122

5.3 Requirements for a Cloud Audit Policy Language . . . 123

5.4 Evaluation of Existing Security Policy Languages . . . 128

5.5 Cloud Audit Policy Language (CAPL) . . . 139

5.6 CAPL Integration into SAaaS Architecture . . . 150

5.7 Evaluation: Cloud Audit Policy Language . . . 154

5.8 Summary . . . 156

6 Anomaly Detection **157**

6.1 Introduction . . . 158

6.1.1 Use Cases for Anomaly Detection in IaaS Clouds . . . 159

6.2 Related Work . . . 160

6.3 Rule Based Anomaly Detection . . . 164

6.3.1 User-centric Anomaly Detection Use Cases . . . 164

6.3.2 Cloud Wide Anomaly Detection Use Cases . . . 165

6.3.3 Rule Based Anomaly Detection Architecture . . . 168

6.4 Behavioural Based Anomaly Detection . . . 171

6.4.1 User-centric Anomaly Detection . . . 177

6.4.2 Cloud Wide Anomaly Detection . . . 183

6.4.3 Anomaly Detection Architecture . . . 187

6.4.4 Neural Networks for Anomaly Detection . . . 194

6.5 Results . . . 199

6.6 Discussions on the Neural Network Approach . . . 204

6.7 Summary . . . 208

7 Security Audit as a Service Prototype 209

7.1 Introduction . . . 210

7.2 The CloudIA Project at Furtwangen University . . . 211

7.3 SAaaS Prototype 1 - Integration of an Agent framework . . . 218

7.3.1 Agent Development . . . 218

7.3.2 SAaaS Graphical User Interface . . . 220

7.3.3 SAaaS Demo 1 - Detection of Cloud Infrastructure Changes . . . 225

7.4 SAaaS Prototype 2 - Integration of the Cloud Audit Policy Language . . . 231

7.4.1 Server and Language Implementation . . . 232

7.4.2 SAaaS Policy Modeller . . . 235

7.4.3 SAaaS Demo 2 - Detection of Scalability Attacks . . . 237

7.5 SAaaS Prototype 3 - Integration of Anomaly Detection . . . 240

7.5.1 Agent Development for Anomaly Detection . . . 241

7.5.2 SAaaS Demo 3a - Login Bruteforce Detection . . . 242

7.5.3 SAaaS Demo 3b - Abnormal VM Creation . . . 245

7.6 Summary . . . 249

8 Evaluation of the SAaaS architecture 251

8.1 Introduction . . . 252

8.2 How SAaaS improves Cloud Audits . . . 252

8.2.1 Cloud Monitoring and Audit . . . 252

8.2.2 Message Reduction and Aggregation of Raw Data at Side to Audit Data by Business Process Awareness . . . 253

8.2.3 Automated Auditing of VM Images . . . 254

8.2.4 The Cloud Audit Test Criteria Catalogue . . . 256

8.3 How SAaaS Addresses Cloud Security Issues . . . 257

8.3.1 Mitigation of Cloud Resources Abuse . . . 257

8.3.2 Mitigation of Shared Technology Issues . . . 258
8.3.3 Better Cloud Monitoring . . . 259
8.3.4 Detection of Account Misuse . . . 261
8.3.5 Detection of Distributed Login Bruteforce Attack . . . 262
8.3.6 Detection of VM Breakout . . . 263
8.3.7 Detection of Cloud Resource Misuse . . . 263
8.4 Summary . . . 264

9 Conclusion & Future Work **267**
9.1 Achievements of the Research . . . 268
9.2 Limitations of the Research . . . 270
9.3 Suggestions & Scope for Future Work . . . 272
9.4 The Future of Cloud Computing Security . . . 273

List of Publications **275**

References **279**

Glossary **299**

Appendix **301**
A.1 DVD ROM Content . . . 303
A.2 Developed SAaaS Agents . . . 305
A.3 Cloud Audit Policy Language Specification . . . 311
A.4 Bootchart Analysis of SAaaS Agents Platform Overhead . . . 323
A.5 Email Correspondence Public Cloud Providers . . . 325
A.6 Simulation sequence scenario “UserAnomaly01” . . . 329
A.7 Conference Poster . . . 331

List of Figures

2.1 Evolution of cloud computing [1, p.4] 13

2.2 Cloud Computing - Technology hype by Gartner [2, 3] 13

2.3 Expected market expansion of cloud computing [4, 5] 14

2.4 Expected business benefits through cloud computing 2010 - 2015 [6] . . 15

2.5 Cloud deployment models [1, p.25] 18

2.6 Cloud service models [1] 20

2.7 The cloud reference architecture based on [7] 22

2.8 Typical components of an IaaS cloud infrastructure 23

2.9 Cloud vendor taxonomy (4th quarter 2011) 29

3.1 Top concerns on cloud computing at organisations [8] 38

3.2 Amazon Health Dashboard accessed at 05/25/2011 and 05/31/2011 . . 39

3.3 Security Risks in a typical IaaS cloud environment 47

3.4 Traditional steps of an IT security audit process 61

3.5 Audit from the Cloud 69

3.6 The cloud reference architecture extended by audit agents 73

3.7 Summary audit test cases “Infrastructure overlapping criteria” [9] . . . 82

3.8 Cutout of audit test criteria catalogue spreadsheet [10] 84

3.9 Audit Test Criteria Catalogue for Cloud Infrastructures [9] categories . 87

4.1 SAaaS event processing sequence 98
4.2 Security Audit as a Service Management Interface 100
4.3 Asynchronous and autonomous agent execution [11] 101
4.4 Agents reduce network load [11] 102
4.5 Types of SAaaS agents distributed in an IaaS Cloud 105
4.6 Basic SAaaS agent design 106
4.7 SAaaS Service Components - User View 108
4.8 SAaaS Service Components - Provider View 109
4.9 SAaaS Agent Platform performance test components: agent deploy time 114
4.10 SAaaS Agent performance test: Agent deploy time 115
4.11 Agent message roundtrip time 115

5.1 Affiliation of cloud Audit Policy Language within SAaaS architecture . 119
5.2 LaSCO example [12] 131
5.3 CAPL class diagram 141
5.4 Integration of CAPL within SAaaS architecture 152

6.1 DDoS Attack Scenario 166
6.2 Distributed SSH bruteforce on weakly secured cloud instances 167
6.3 User-centric anomaly detection: cloud scalability 169
6.4 Cloud wide anomaly detection: Login bruteforce detection 170
6.5 Anomaly Detection: online movie rental - normal behaviour 174
6.6 Anomaly Detection: online movie rental - abnormal behaviour 175
6.7 Calculation of a VM's runtime 179
6.8 User Anomaly - Normal Behaviour time over training days 181
6.9 User-centric anomaly detection - Normal user behaviour 182

6.10 CloudWideAnomaly01 scenario data set 186
6.11 UserAnomaly01 scenario data set 189
6.12 UserAnomaly02 scenario data set 190
6.13 UserAnomaly03 scenario data set 190
6.14 Cloud usage data simulator 191
6.15 Anomaly detection evaluation 194
6.16 Feed forward net for scenario “UserAnomaly01” 195
6.17 Feed forward net for scenario “UserAnomaly02” and “UserAnomaly03” 197
6.18 Neural networks learning curve in scenario UserAnomaly03 198
6.19 Feed forward net for scenario “CloudWideAnomaly” 198
6.20 Anomaly Detection - simulation results (UserAnomaly03) 200
6.21 Anomaly detection - weighted input classifier (UserAnomaly01) 200
6.22 Anomaly detection - weighted input classifier (UserAnomaly03) 201
6.23 Anomaly Detection - simulation results (ProviderAnomaly01) 202

7.1 Cloud Infrastructure and Applications architecture [13] 211
7.2 Cloud Infrastructure and Applications architecture module overview . . 212
7.3 CloudIA hardware infrastructure at Furtwangen University 215
7.4 CloudIA software components 216
7.5 SAaaS prototype components - version 1 219
7.6 SAaaS prototype version 1 - Agents deployed 220
7.7 SAaaS prototype - SAaaS agents 222
7.8 SAaaS prototype: SAaaS GUI elements 223
7.9 SAaaS prototype: graphical agent management 223
7.10 SAaaS prototype: Security & Audit Dashboard 224
7.11 SAaaS prototype: SAaaS Provider Dashboard 224
7.12 SAaaS Demo 1 - Detection of cloud infrastructure changes 226

7.13 Assignment of security policy template for web server protection 227
7.14 Cloud security dashboard prototype before and after detected attack . 227
7.15 SAaaS prototype: Demo 1b - concurrent audits 228
7.16 SAaaS prototype: Demo 1b Audit agent arrived at target VM 229
7.17 SAaaS prototype: Demo 1b graphical log of demo sequence 230
7.18 SAaaS prototype components - version 2 231
7.19 SAaaS prototype 2 - graphical policy modeller 232
7.20 CAPL Database Design 234
7.21 SAaaS Policy Modeller: Create Policy 235
7.22 SAaaS Policy Modeller: Manage existing policies 236
7.23 SAaaS Policy Modeller: Group creation and management 237
7.24 SAaaS cloud GUI: Web server VMs created 238
7.25 Cloud security dashboard: Mitigated scalability attack 239
7.26 SAaaS prototype components - version 3 240
7.27 Anomaly warning in SAaaS Dashboard 241
7.28 SAaaS prototype demo 3a: Detected abnormal login 243
7.29 SAaaS prototype demo 3a: Detected abnormal login 245
7.30 SAaaS prototype demo 3b: Anomaly detection for VM life cycle behaviour 246
7.31 SAaaS prototype demo 3b: Abnormal VM creation detected 247
7.32 SAaaS prototype demo 3b: Anomaly server output 247
7.33 SAaaS Prototype 3 - Training of user behaviour 248

8.1 Distributed Denial of Service attack from the Cloud 264

A.1 CAPL class diagram 311
A.2 SAaaS Demo2: Concurrent cloud audit in case of upscale event 322
A.3 Bootchart analysis of VM without SAaaS agent platform 323
A.4 Bootchart analysis of VM with SAaaS agent platform 324

List of Tables

2.1 Roles in cloud computing 24

3.1 What the cloud provider controls [14] 40
3.2 Overview of Hypervisor outbreaks [15] 51
3.3 Overview of cloud security problems 59
3.4 Industry standards for IT security [16] 66
3.5 Virtual Appliance Audit Categories 79

4.1 Evaluation: SAaaS agent performance lab setup 113

5.2 REI characteristics 129
5.3 Ponder characteristics 130
5.4 LaSCO characteristics 132
5.5 CIM & CIM policy model characteristics (version 2.34) 134
5.6 CIMI characteristics (version 1.0.1) 136
5.7 Comparison of existing security policy languages 138
5.8 CAPL data types 142
5.9 CAPL classes: basic attribute structure 143
5.10 CAPL class: cloud Entry Point 144

5.11 CAPL class: AbstractCollection . . . 145
5.12 CAPL class: Machine . . . 145
5.13 CAPL class: MachineCollection . . . 146
5.14 CAPL class: Group . . . 146
5.15 CAPL class: RuleTypes . . . 147
5.16 CAPL class: Policies . . . 148
5.17 CAPL class: PolicySet . . . 149
5.18 CAPL class: Action . . . 149
5.19 Evaluation of CAPL . . . 155

6.1 Considered cloud security risks . . . 161
6.2 Anomaly detection: direct input parameters for IaaS cloud usage . . . 173
6.3 Anomaly detection: feature for IaaS Clouds usage . . . 176
6.4 Example data set, normal behaviour: one user, one VM over ten training days . . . 182
6.5 Anomaly detection - simulation environment . . . 199
6.6 Results of scenario Useranomaly03 . . . 201
6.7 Anomaly Detection - simulation results (ProviderAnomaly01) . . . 203

7.1 Research projects build on CloudIA architecture . . . 213
7.2 SAaaS agents developed during 1st SAaaS prototype stage . . . 221
7.3 SAaaS agents developed during 2nd SAaaS prototype stage . . . 237
7.4 SAaaS agents developed during 3rd SAaaS prototype stage . . . 242

8.1 Cloud Security Issues addressed by SAaaS . . . 261

A.1 CAPL classes: basic attribute structure . . . 312
A.2 CAPL class: Cloud Entry Point . . . 313
A.3 CAPL class: AbstractCollection . . . 313

A.4 CAPL class: Machine 314
A.5 CAPL class: MachineCollection 314
A.6 CAPL class: MachineTemplate 315
A.7 CAPL class: MachineTemplateCollection 315
A.8 CAPL class: Group 315
A.9 CAPL class: GroupCollection 316
A.10 CAPL class: RuleTypes 316
A.11 CAPL class: RuleTypeCollection 317
A.12 CAPL class: Policies 317
A.13 CAPL class: PolicyCollection 318
A.14 CAPL class: PolicySet 318
A.15 CAPL class: Action 319
A.16 CAPL server: Implemented HTTP return values 319

Acknowledgements

This work was made possible due to funding from the Federal Ministry of Education and Research of Germany and the Faculty of Computer Science at Furtwangen University. I wish to thank both organisations for their support. I'd like to thank all team members at the Cloud Research Lab and Plymouth CSCAN group for their support, especially for asking critical questions at the right time and their feedback.

The work, and indeed this PhD, would not have been possible without the help and support of my Director of Studies, Dr. Christoph Reich. Thanks go to him for his tireless effort in steering me through the PhD process, from publishing papers to presenting at international conferences. His professionalism and experience has been invaluable, and I owe much of my success to his guidance. It is said, a teacher is immortal. Even when he leaves his body, the values he taught live on within the heart of his students. I am grateful for his support and proud that he became supervisor, respected colleague and dear friend to me.

Thanks also go to my supervisors, Dr. Nathan Clarke and Dr. Martin Knahl, who have spent a lot time proof reading papers and my thesis, in addition to providing helpful experience and guidance throughout my studies. Without the help of Dr. Clarke I would not have been able to tackle down the problem of understanding classifiers in neural networks! Additionally, his response time in reviewing papers is outstanding.

I would like to thank my colleagues Dr. Anthony Sulistio, Hendrik Kuijs and Thomas Ruebsamen for the excellent working atmosphere that really counteracted a lot of stressful situations and thus helped to get through some rather difficult times. Also I'd like to thank Benjamin Sanders and Nina Baer for their support and shelter during my time in Plymouth!

The work of this thesis was supported by multiple thesis students, who helped me with the little nasty bits and pieces - from implementing software agents, preparing conference demonstrations and first level support when a demo was not working ten minutes prior to a presentation at the other side of the world. Explicitly I'd like to thank Christian Fischer, Thomas Ruebsamen, and Florian Kemmer for their always-on support. I am very proud of their development, and grateful that I was able to be a part of it.

I'd like to thank my friends Markus, Eli, Javier, Henriette and Daniel for providing me with support by believing in me, enhancing my mathematical models, sharing exciting moments or simply for being there.

A very special "thank you" goes to my brother Stefan, my Mum Angelika and my Dad Martin for supporting me through my entire life.

Authors Declaration

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other University award.

This study was financed with the aid of a research grant from the Federal Ministry of Education and Research (BMBF) of Germany.

Relevant seminars and conferences were regularly attended at which work was often presented and several papers prepared for publication, details of which are listed in the appendices.

Word count of this PhD thesis: 61.227
The term cloud has been used: 1899 times

Signed .....

Date .....

1 Introduction & Overview

While 80% of IT decision makers of Fortune 500 companies recognised cloud technology as giving their organisation a competitive advantage, security is still a major concern and barrier to cloud adoption [17]. It outranks even compliance and integrity. Thus, there is a high need for developing a cloud security solution. This research presents a novel approach of mitigating cloud specific security challenges with targeted concurrent cloud audits.

1.1 Introduction

Growing network and therefore internet bandwidth in combination with virtualization technology lead to a new form of utility computing, which allows us to use computing resources in a flexible and on-demand fashion. Within the last four years, cloud computing changed information technology drastically in multiple application areas. Whereas USB sticks or external hard disk were the common way of quickly exchanging large data between two parties, such as a photo collection or large simulation results, cloud storage services like Dropbox [18], ownCloud [19] or Apple iCloud [20] replaced them. Start-up companies don't invest a high amount of hardware for their first data centre, but renting Software as a Service products, such as Google Mail for email services, Office 365 [21] or Google Apps [22] for office applications. If computing power is needed for customised applications, public cloud infrastructure provider, such as Amazon Web Services [23] offer virtual PCs on-demand on a pay per use model for just a couple of cents per hour [24]. Companies can focus on the development of their real products, instead of taking care of data centre design, acquisition of hardware and training of administration personell. Industry is taking advantages of this new, flexible available on-demand cloud computing model by outsourcing services to cloud service providers (CSP), or using the technology in their own data centres to increase flexibility and accelerate deployment of IT resources. The convenience, efficiency and cost benefits of cloud computing has even made hospitals to move health care data and records to the cloud [25].

This lets assume, that security is one of the major targets of cloud providers offering cloud computing services. However, missing security standards, numerous security incidents at public cloud infrastructures as well as a rather reserved information policy on security measures taken to protect customer data, has lead to become cloud

computing security the number one concern of IT decision makers, when it comes to utilising cloud services [8]. In case, data stored in cloud resources gets stolen or becomes unavailable for multiple days due to a failing network connection, the financial loss to the user in this case would not only be that one particular service is not available, but could lead to a complete threat of the companies existence, since suddenly no IT services are available anymore. Thus, a detailed analysis of cloud computing security risks, possible mitigation approaches and enhanced cloud security systems is necessary to establish trust in the technology and increase security.

1.2 Aims & Objectives

Literature review on related work in the area of cloud computing security shows, that the following limitations exist:

- Existing security guidelines and frameworks stay rather high level, lacking the necessary technical depth and needs, especially for IaaS clouds
- SaaS cloud audits exist, however IaaS cloud audits are missing
- A cloud and hypervisor independent audit system for IaaS clouds is missing
- Existing cloud audits require specials hardware
- A security language for IaaS cloud audits is missing. Existing security languages are not applicable
- No cloud data set including cloud specific attacks is freely available

Thus, this research on security audit compliance for cloud computing develops a novel cloud auditing system: the Security Audit as a Service architecture. It addresses the identified limitations by providing the following features:

- A detailed IaaS cloud audit criteria catalogue with sufficient technical depth for cloud customers and providers

- A technology independent, cloud provider interoperable audit system for IaaS clouds, which does not require special hardware
- A hypervisor technology independent sensor system utilising software agents
- A security language to model IaaS cloud security demands and corresponding audit checks
- Establishment of contact with cloud providers to build a cloud usage data set including cloud attacks or development of a cloud data set simulator

The aim of this research is to identify cloud specific security challenges in Infrastructure as a Service (IaaS) clouds and how they can be addressed. This has been achieved by developing a flexible, on-demand cloud audit system, which performs specific targeted security audits in case the cloud infrastructure changes. In order to achieve this, the research can be divided into six distinct phases.

- To identify differences between traditional and new, cloud specific security challenges for IaaS clouds
- To investigate how security audits need to change, to address the identified cloud specific security challenges
- To design and evaluate a new cloud security audit architecture, which aims to increase transparency for cloud consumers and cloud providers
- To design a cloud audit policy language to model desired security states of cloud components and corresponding cloud audits to validate these states.
- To design and elaborate that an anomaly detection system can identify advanced cloud specific security challenges by utilising machine learning techniques
- To implement and test a prototype of the cloud audit system to demonstrate its practical effectiveness

The first part of the research contains a comprehensive review and discussion about differences between traditional - but through cloud characteristics enhanced - security challenges and cloud specific security challenges. Through an understanding of the development of cloud computing technology and their specific characteristics, this phase of the research provides the comprehensive arguments and the basis for the need of a cloud specific security audit system. This identified, the second phase of the research proceeds with an analysis of security audits, giving insights on how they need to change to address the identified cloud security challenges. The outcome of this phase is an audit criteria test catalogue for cloud infrastructures, containing over 140 security checks for cloud customer and provider.

In order to achieve the objective of transparently utilising cloud audits to address cloud specific security challenges, it was imperative to design an on-demand audit architecture for IaaS clouds. From the established requirements, the approach of utilising lightweight software agents to perform specific targeted so called "concurrent" security audits of cloud instances proved to be feasible. However, a security policy language was needed to design the desired security states of cloud instances and corresponding cloud audits and agents to monitor and evaluate this security state. Therefore, during the next phase of research a comprehensive analysis on existing security policy languages was undertaken. Since none of the existing languages fulfilled the established requirements, a new Cloud Audit Policy Language (CAPL) was developed. To address more advanced cloud specific security challenges, the research showed that two different classes of mitigation approaches prove promising: rule based anomaly detection and behavioural based anomaly detection. Whereas rule based anomaly detection proved to be already possible with the developed audit architecture and cloud audit policy language from earlier research phases, deeper research on pattern matching and machine learning technique was performed to address the class of behavioural based anomaly detection in clouds. As a result, an anomaly detection system, based on neu-

ral networks proved to successfully identify advanced cloud security challenges and promises high potential on mitigation of future cloud security problems arising from misuse of cloud environments. The completion of the final phase was to design and evaluate a prototype of the developed Security Audit as a Service architecture, which includes a prototype of the Cloud Audit Policy Language and the designed anomaly detection system. Three demonstrations were built, to evaluate the effectiveness of the developed approach.

1.3 Thesis Structure

The thesis addresses the aforementioned objectives in order and is comprised of the following Chapters.

Chapter 2 - Review of Cloud Computing, starts with an introduction into cloud computing technology. A brief evolution is presented and key definitions are established. Cloud computing consists of different deploy and service models, which get elaborated. Following on, a cloud computing reference architecture and typical participating roles of cloud players gets presented. After discussing related technologies to cloud computing, the commercial cloud landscape is drawn. The Chapter finishes with a discussion on advantages and disadvantages of cloud computing.

Chapter 3 - Cloud Infrastructure Audit presents the first main stage of research - the identification of cloud specific security issues. It shows, how cloud security audits need to adapt to be an adequate measure to mitigate the presented security issues. As a result of this part of the research, a Cloud Audit Test Criteria catalogue gets presented, which is targeted for the cloud user as well as the cloud provider to get a transparent knowledge of an IaaS cloud's security status.

After laying out the ground of requirements for cloud audits, *Chapter 4 - Security Audit as a Service* presents a cloud audit architecture to address the presented cloud security issues. First, the context is set by definition of involved users and groups. Then, the Security Audit as a Service (SAaaS) architecture gets presented. It is based on lightweight software agents, which perform targeted audits in case of a cloud infrastructure change. Thus, a detailed description on the developed agents is given. The Chapter concludes with a presentation of all developed SAaaS services. The SAaaS architecture forms the second stage of the main research, which gets extended by the work of the following Chapters.

To execute specific targeted audits, business processes of cloud components as well as a desired security status of the virtual instances needs to be definable. Thus, *Chapter 5 - A Cloud Audit Policy Language* presents the third part of the main research. First, the context is set by a presentation of use cases and the definition of requirements on a cloud audit policy language. Based upon those, a comparison of existing security languages is provided. Since no existing language satisfied the established requirements, the Chapter continues with a detailed presentation of the developed Cloud Audit Policy Language. The Chapter finishes with a description of how the Cloud Audit Policy Language integrates into the SAaaS architecture, forming its second stage of expansion.

To address advanced and future cloud security issues, *Chapter 6 - Anomaly Detection in IaaS Clouds* presents a cloud anomaly detection system. After setting the context, a distinction between rule based and behavioural based detection methods is made. First, rule based anomaly detection gets elaborated, followed by a behavioural based detection system. This utilises a machine learning approach for advanced cloud misusage detection. To show feasibility of the system, a cloud usage data simulator gets

presented, simulating selected user specific and cloud wide anomaly scenarios. Results are presented and discussed, before the Chapter closes. The anomaly detection system forms the third development stage of the SAaaS architecture.

To practically validate the benefit of the whole research, a prototype of the developed cloud audit system gets presented in *Chapter 7 - Security Audit as a Service Prototype*. The Chapter follows the structure of the main research phases and presents the three consecutive development stages of the SAaaS prototype:

- Stage 1 - SAaaS architecture with agents performing concurrent cloud audits
- Stage 2 - Integration of the Cloud Audit Policy Language
- Stage 3 - Integration of an anomaly detection system for IaaS clouds

Stage one, starts with the basic architecture development, which focuses on the concept of using agents to perform specific targeted cloud audits. For the second prototype stage, integration of a cloud policy language gets discussed. The third extension of the prototype introduces the integration of the rule based and behavioural based anomaly detection system into the SAaaS prototype. Each prototype Section ends with a demonstration of this specific part of the development stage.

Chapter 8 - Evaluation of the SAaaS Architecture evaluates the presented research against the established issues. Therefore, it discusses first how the developed SAaaS system addresses the presented cloud specific security issues and how cloud audits are improved by it. Then, a technical evaluation on the performance of the software agents is presented, followed by a comparison Cloud Audit Policy Language against the established requirements. The Chapter finishes with a discussion on the developed anomaly detection system and justifies the chosen approach of simulating cloud usage data and using neural networks as a machine learning technique.

Finally, Chapter 9 presents the main conclusions arising from the research, highlighting its key achievements and limitations. It also contains a discussion on areas for future research and development. The thesis also provides a number of appendices in support of the main discussion, including the Cloud Audit Test Criteria catalogue, a detailed overview of developed software agents, the CAPL language specification and more technical information. The appendices also contain 15 published papers arising from the research programme.

Each Chapter starts with an introduction into the corresponding research area. The goal of the research subtopic is formulated (heading "Summary of Research"). Related work of each topic is presented at the beginning of each chapter, after use cases or requirements have been established. After presenting the research, every chapter ends with a summary of the work presented.

Nomenclature

Within this thesis, cloud computing related terms are used. Technical terms are introduced at their first appearance, a collection of the most important ones are listed in Section 9.4 - Glossary. The terms cloud customer, cloud consumer and cloud user are used interchangeably throughout this work.

1.4 Security Audit as a Service Research Grant

The research topic "Security Audit as a Service" presented in this thesis, was applied for a research grant at the program "Information Society - IT Security" at the Federal Ministry of Education and Research (BMBF) of Germany. The program committee found the research on cloud security issues and the proposed Security Audit as a Service system eligible and issued a grant for the period of two years, starting June 2011. It is referenced under the grant number 01BY1116.

2 Review of Cloud Computing

"I think there is a world market for about five computers."

(Thomas J. Watson, Chairman of the Board of International Business Machines (IBM), 1943)

This chapter introduces the concept of cloud computing by presenting its evolution as a service computing methodology and elaborating necessary basic knowledge on the technology, which is necessary to follow the research presented in this thesis.

2.1 Introduction

Cloud computing represents one of the most significant shifts in information technology many of us are likely to see in our lifetimes [26]. This chapter will show the evolution of cloud computing and why it is such a relevant topic in information technology right now. The introduction of important cloud deployment and service models as well as a presentation of a cloud reference architecture lays out the technical foundation for this thesis. Typical roles and related technologies are elaborated and an overview about the current provider landscape is given. Parts of this Chapter's work have been published in "Cloud Infrastructure & Applications - CloudIA" [27].

2.2 Evolution of Cloud Computing

Cloud computing is often described as the new computing paradigm, which will completely change the consumption of computing. Nicolas Carr describes cloud computing in "The Big Switch" [28] as an information revolution within the information age and compares it with a major development of the industrial era: the introduction of electricity. Mather et al. say in "Cloud Security and Privacy", that "... Cloud Computing itself is a logical evolution of computing." [1, p.3]. Figure 2.1 shows cloud computing as a development in the Internet service provider (ISP) model. While the first ISPs (ISP 1.0) just provided Internet access, their business model developed further to offering general services like access emails and servers (ISP 2.0). Enterprises asked for special services and ISPs offered specialised data centers for hosting customers' servers including the infrastructure needed to run them, which is defined as "co-location facilities" (ISP 3.0). The successor of this were application service provider (ASPs), which offered not just bare computing infrastructure but higher services like customised application for organisations (ISP 4.0). The main difference between ASPs and cloud

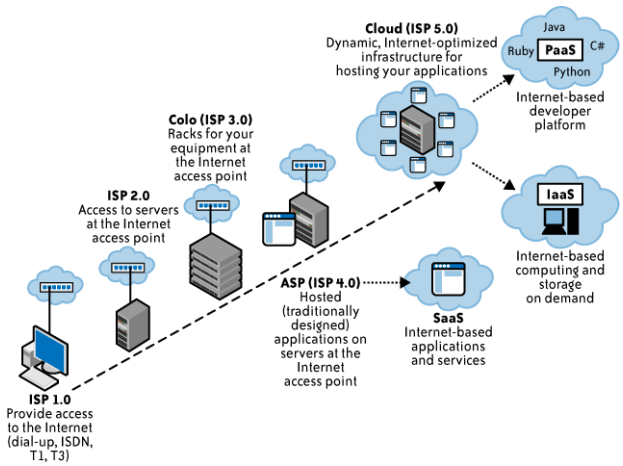
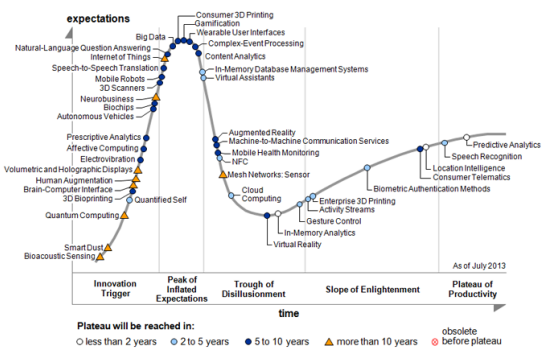
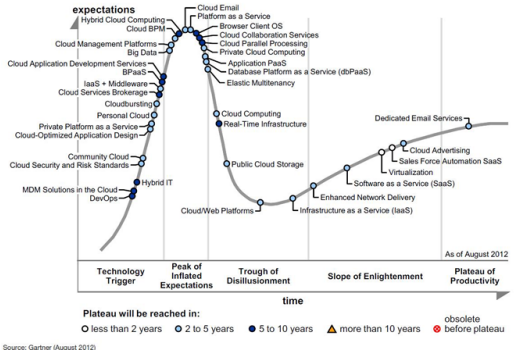


Figure 2.1: Evolution of cloud computing [1, p.4]

computing providers is the underlying infrastructure design. ASPs already offered a specific service to multiple customers, but commonly through dedicated infrastructures, meaning each customer used its own dedicated instance, which usually ran on a dedicated server and no other customers used the service on this particular server. In cloud computing, especially in a Software as a Service (SaaS) model, services are offered on infrastructure, shared by multiple customers [1, p.3,4]. The Gartner Research Group yearly investigates new technologies and how to discern a hype from what's commercially viable [29], see Figure 2.2. While all cloud technologies were summarised under the term "The Cloud" in the 2009 hype cycle, it was divided in



(a) Technology hype cycle 2013

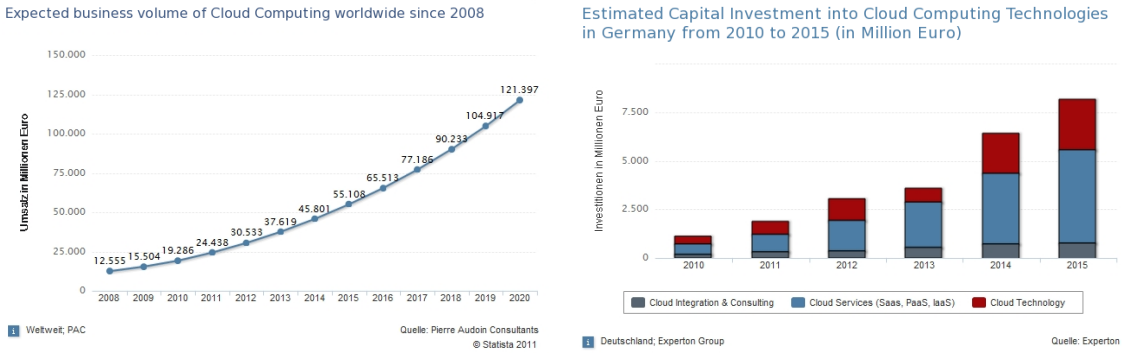


(b) Cloud Computing specific hype cycle 2012

Figure 2.2: Cloud Computing - Technology hype by Gartner [2, 3]

three entries: Cloud Computing, Cloud/Web Platforms and Private Cloud Computing in 2010. In 2013, Gartner rated cloud computing to be over the status of being just a hype (Figure 2.2(a)), getting closer to productivity as compared to 2009. Since 2011, Gartner devotes a complete own hype cycle to cloud computing, Figure 2.2(b) shows the last version from 2012. Gartner sees Cloud/Web Platforms even closer to productivity than one year before. Furthermore, most of the cloud technologies were classified 2-5 years or 5-10 years from maturity, which is also an indicator of the increasing importance of cloud computing for information and computing technologies. In 2013, Gartner's analysts expect IT decision makers now to have more realistic strategies for integrating this technology, due to not accomplished cost advantages in former years [30].

Industry analysts are foreseeing a huge financial potential in cloud computing, as Figure 2.3 shows. A business volume of 121,897 million Euro for cloud computing is expected (Figure 2.3(a)) until 2020. The estimated capital investment in cloud computing in Germany alone is expected to rise by 871% (see Figure 2.3(b)) from 548 million Euro in 2010 to 8.775 million Euro by 2015 [5]. As depicted in Figure 2.4, high economic benefits are expected from cloud computing to business and the wider EMEA economy of France, Germany, Italy, Spain and the UK [6].



(a) Expected business volume of cloud computing 2008-2020

(b) Expected capital investment cloud computing in Germany

Figure 2.3: Expected market expansion of cloud computing [4, 5]

The study states, that across the five economies as a whole, widespread adoption of cloud computing has the potential to generate over 763 billion Euro of cumulative economic benefits over the period 2010 to 2015. This is 1.57% of the total estimated cumulative GDP of the five economies over the same period [6, p.7]. Furthermore, it states that cloud computing could improve the efficiency of an average employee by 2.1%, while also reducing the amount of investment tied up in underutilised IT capacity.

EMEA: Cumulative Economic Benefits 2010-2015

	France	Germany	Italy	Spain	UK	EMEA
	€ mil	€ mil	€ mil	€ mil	€ mil	€ mil
Business development opportunities	24,599	32,642	23,995	16,866	29,555	127,657
Business creation	51,377	69,507	43,305	30,939	20,026	215,153
Net total cost savings *of which:*	26,323	37,740	28,463	22,008	26,206	140,740
– IT CapEx savings	*28,653*	*36,378*	*30,461*	*23,013*	*36,176*	*154,682*
– IT OpEx savings (FTEs / productivity)	*13,818*	*18,139*	*14,533*	*10,396*	*16,943*	*73,829*
– IT OpEx savings (power & cooling)	*11,107*	*14,345*	*11,821*	*8,510*	*10,566*	*56,349*
– additional cloud services expenditure (PAYG) *	- *27,255*	- *31,122*	- *28,353*	- *19,910*	- *37,481*	- *144,120*
Indirect GVA	60,450	81,351	55,007	40,737	42,202	279,747
Total Economic Benefit	**162,749**	**221,239**	**150,770**	**110,550**	**117,989**	**763,297**
Direct and Indirect employment ('000s)	**469.4**	**789.4**	**455.8**	**392.5**	**289.0**	**2,396.2**

** This category of spend is captured for private cloud through lower firm-level CapEx and OpEx savings. The firm-level CapEx and OpEx savings are higher under hybrid and public cloud, but there can expect to also be incurred new spend on external cloud services*

Source: Cebr analysis

Figure 2.4: Expected business benefits through cloud computing 2010 - 2015 [6]

2.3 Definition of Cloud Computing

The key of cloud computing lies in its component-based nature, such as reusability, substitutability (e.g. alternative implementations, specialised interfaces and runtime component replacements), extensibility, customizability and scalability [31]. Foster et al. [32] discuss the basic concepts of cloud computing and identify the differences compared to grid computing. In addition, Armbrust et al. [33] give a good overview of cloud computing by highlighting some of the obstacles and opportunities in the field. The definition of the US National Institute of Standards and Technology (NIST) has

evolved into a de facto standard for cloud computing: *"Cloud Computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three service models and four deployment models"* [34].

NIST's definition identifies the following main cloud characteristics:

On-demand self service Cloud customer can provision and manage computing power and network storage without any human interaction with a service provider.

Broad network access Cloud resources are accessed via the network (mostly the Internet) using standardized Internet protocols.

Resource pooling A providers' computing and storage resources are shared between multiple customers as a multi-tenant model. A customer has no control or knowledge over the exact physical location, where its data is stored, or where its rented resources are executed.

Rapid elasticity Resources can be deployed elastically and scaled rapidly to fulfil the current demand by scaling up and down automatically. The cloud customer only pays for the resources and services they actually use.

Measured service Resources consumption gets measured and optimised by cloud control systems depending on how each customer uses its cloud resources (storage, processing, bandwidth). Transparency is provided to the provider and its customers by monitoring and reporting of the cloud usage.

Cloud computing is not a new technology, in fact it combines known and established technologies, such as virtualization and infrastructure management to provide IT ser-

vices as an on-demand model. In this work it is defined that cloud computing delivers Infrastructure-, Platform-, and Software as a Service (IaaS, PaaS, and SaaS) on a simple pay-per-use basis.

2.4 Cloud Deployment & Service Models

NISTs' definition of cloud computing [35] states four deployment models, see Figure 2.5, which are typical for cloud computing: Private Cloud, Community Cloud, Public Cloud and Hybrid Cloud.

Public Clouds - also known as *external Clouds*, are operated and managed by a third-party vendor (business, academic or government) for open use by a non-limited group of customers. Services are offered over the Internet and accessible through web applications, web services or established data communication protocols like Secure Shell (SSH). Security management is done by the vendor, responsible for the public cloud offering. Therefore, customers don't have a good insight into physical and logical security measures of the private Cloud [1, p.23]. A popular commercial public cloud offer is the Elastic Cloud (EC2) by Amazon Web Services (AWS) [23].

Community Clouds - the cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organisations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations) [35, p.3]. Typically it is owned and managed by one or multiple members of that community, who are also responsible for the security management of the cloud environment. G-Cloud [36] - a cloud environment providing UK governmental agencies with information and communications technology (ICT) services is an example of a community cloud.

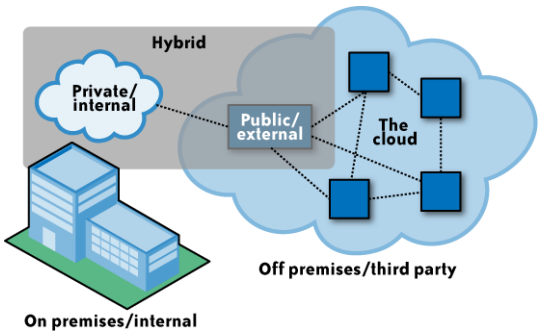


Figure 2.5: Cloud deployment models [1, p.25]

Private Clouds - also known as *internal Clouds*, are cloud offerings exclusively used by a single organisation, such as a large enterprise serving its internal customers, such as different branches. The cloud infrastructure is commonly run on private networks, operated and managed by the organisation itself, a third party or a combination of both [35, p.3]. Cloud Infrastructure and Applications (CloudIA) - a cloud environment for students at the University of Applied Sciences Furtwangen (HFU) is an example of a private cloud.

Hybrid Clouds - are a combination of the private cloud and the public cloud model. In a hybrid cloud, both clouds exist independently including their corresponding characteristics. Users are consuming most of the time the Private Cloud resources, but in specific situations, such as load peeks resources from the public cloud are used to satisfy the demand. This special case is also known as "Cloud Bursting". It ensures, that a sudden increase in computing requirement is handled gracefully. In a hybrid cloud scenario, sensitive data is commonly kept in the private cloud, and only non-sensitive data gets processed in the public cloud. An example for a hybrid cloud scenario is the combination of a private cloud infrastructure and a security vendor's public network of threat intelligence delivered through public cloud services, such as Trend Micro's Smart Protection Network [37].

Cloud Service Models

The following three fundamental service models can be identified in cloud computing, which also correlate to NIST's cloud definition: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). They describe the degree of a cloud's availability to users.

Infrastructure as a Service

This layer provides an abstract view on the underlying hardware, which includes PCs, mass storage systems or network components. They get logically partitioned and provided as virtual resources, by utilising virtualization technology such as, KVM [38], Xen [39], VMware [40]. Thus, a flexible management of the provided resources is possible, allowing a cloud user to create, start, stop, delete or scale virtual resources. In contrast to traditional IT-Hosting services, only a virtual instance gets provided instead the whole physical machine. This enables CSPs to optimally divide available hardware resources and provide customers with an on-demand self service on a pay-as-you-go basis. In IaaS, cloud service provider only ensure availability and usability of the infrastructure. Management of rented virtual resources, installation of additional services, such as web server or email server as well as connectivity to externals systems are in the responsibility of the cloud customer. In some cases, where the operating system is included and a software license is required the software license costs are either amalgamated into the costs for the service or included as a surcharge [41]. A special sub-model of IaaS is the Storage as a Service model, where only storage space is provided by the CSP. Popular IaaS providers are Amazon Web Services [23], Rackspace [42] or Hosting.com [43]. The currently most popular storage as a service provider is Dropbox [18]. The research, presented in this thesis is residing on the IaaS layer.

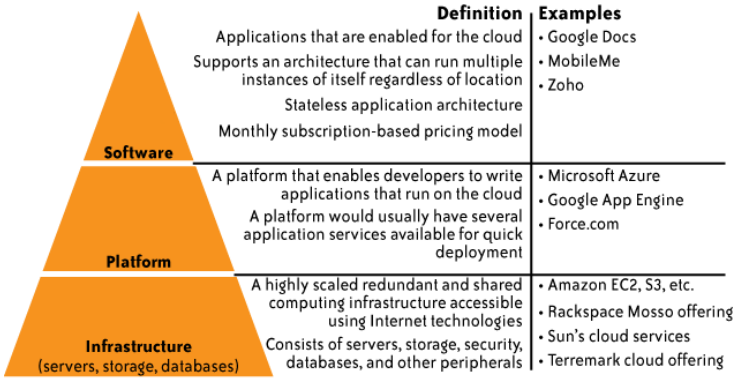


Figure 2.6: Cloud service models [1]

Platform as a Service

Platform as a Services delivers a comprehensive development environment to application developers. The developed applications are also offered through the PaaS platform [1]. PaaS customer don't have to deal with installation and configuration of a virtual server, since this is provided by the CSP. The provider defines the programming languages, which can be used and supplies toolkits, developer standards, libraries, databases, a software development environment and APIs as well as distribution- and payment channels. This enables a multi-tenant application development with a rapid application propagation with minimal entry costs. In a complete PaaS solution, developers can build and deploy web applications without installing any tools on their computer. Popular PaaS offers are Google Apps [22], Force.com [44] and Microsoft Windows Azure [45].

Software as a Service

In the Software as a Service model, the cloud provider also acts as an application developer. The cloud customer rents software for use on a subscription or a pay-per-use basis. It is mostly used through a web browser or an adapted interface provided by the cloud service provider (CSP). The main difference between the traditional software

model and the Software as a Service model is its multi-tenancy orientation. Software is not just purchased and installed on one specific server, serving one single customer's end user group. The SaaS model is a multi-tenant architecture, which means the physical backend is shared among different customers but is logically unique for each customer. It is common that SaaS provider utilise PaaS or IaaS provider to host their services, to benefit from scalability advantages. Popular SaaS providers are Google Docs [46], Microsoft Office Online Services 365 [21] and Salesforce.com [47]. Figure 2.6 depicts the introduced service models and lists examples of commercial cloud provider for the corresponding service.

2.5 Cloud Computing Architecture

Each cloud architecture includes certain components, independently of the targeted cloud service model. Figure 2.7 shows a cloud computing reference architecture, described by Grobauer et al. [7], which is based on research by the University of California, Los Angeles and IBM [48]. This architecture serves as a fundamental basis architecture of this research, and gets enhanced during the presentation of this dissertation. It includes the most security-relevant cloud service components and shows, which of them are managed by the cloud provider. Commonly, the provider maintains one or multiple data centers (facilities) including necessary utilities, such as electricity or air condition. In this facilities hardware is operated, which are typically physical cloud hosts as well as networking equipment such as, switches, routers and network cables. On a cloud host, an Operating System (OS) and applications supply basic supporting functionality, like network connectivity or inter-process communication. On top of that, the hypervisor layer provides the ability of flexible resource pooling. The cloud software infrastructure and cloud software environment are an abstraction of the cloud resources provided as a service including all necessary software components, such

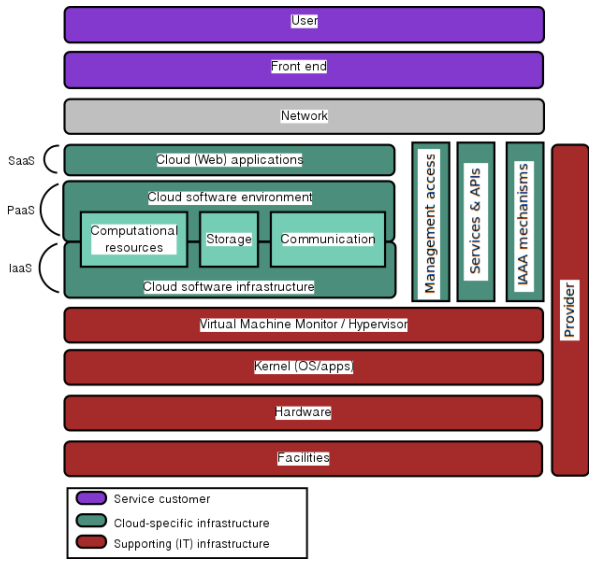


Figure 2.7: The cloud reference architecture based on [7]

as a cloud management software. It interacts with supporting services like Domain Name System (DNS), Management access systems, e.g., an admin management web console, and identity, authentication, authorisation and auditing mechanisms (IAAA). On top of that is the cloud web application layer, which typically uses a browser front end for user interaction. Up to this layer, depending on the cloud service model, the service components are controlled by the provider. On top of that, the network layer gets explicitly pointed out, since this is the network between the cloud provider and the cloud customer, and is mostly fully untrusted (Public Cloud) or semi trusted (Hybrid Cloud, Private Cloud), depending on the cloud deployment model.

Since this research is focused on Infrastructure as a Service Clouds, Figure 2.8 shows a high level overview of typical technical infrastructure components of an IaaS Cloud. User access is typically provided through numerous client devices using standardized network protocols, such as SSH or HTTP. Furthermore, an application programming interface (API) is provided, using Representational State Transfer (REST) or XML Remote Procedure Calls (RPC). As a core component within an IaaS Cloud, the

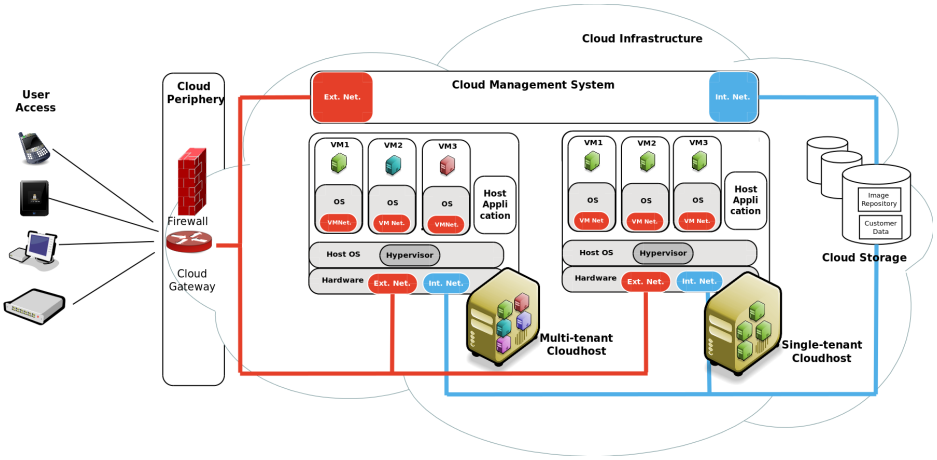


Figure 2.8: Typical components of an IaaS cloud infrastructure

cloud management system manages access and deployment of cloud resources. Cloud instance, such as virtual machines (IaaS), customer developed web applications (PaaS) or rented software instances (SaaS), are deployed and executed on cloud hosts. Data, such as VM images or customer data are stored on a cloud storage system. Typically, a distributed filesystem is used for this and copies are distributed over multiple storage components. Virtual machines are executed on a cloud host through a virtualization hypervisor, such as Xen, KVM or VMware. Dependent on the service model, customer share a physical cloud host (multi-tenancy) or use a customer exclusive cloud host (single-tenancy). Data centre peripherals, such as firewalls, routers, switches, and network connections are extended by virtualized components and supplied by the cloud service provider.

2.6 Roles in Cloud Computing

The cloud computing paradigm mainly identifies three roles: provider, integrator and consumer. Each of these roles include specific responsibilities and expectations relative to another [41]. An investigation among different cloud service providers was done during the initial phase of the research. Depending on the considered service

Cloud Service Model	Identified Roles
*	Cloud service assembler
IaaS	Cloud provider, cloud customer
PaaS	Cloud provider, service provider, cloud customer, service developer, user / user groups
SaaS	Cloud provider, service provider, cloud customer, service developer, user / user groups

Table 2.1: Roles in cloud computing

model (IaaS, PaaS, SaaS), it can be differentiated between multiple roles, which will be elaborated in the following. Table 2.1 summarizes the identified roles over the common cloud service models.

IaaS roles

In an IaaS scenario two roles can be identified: *cloud provider* and *cloud customer*. The *cloud provider* supplies and manages the infrastructure to run virtual machines. He enables cloud customers to access the virtual machines, commonly through SSH (Linux/Unix VMs) or Remote Desktop Protocol (RDP) (Windows VMs) connections. Typically, the provider provides preconfigured virtual machine images. The customer is exclusively responsible for the software stack inside a deployed VM. The *Cloud customer* is a person or organisation, which runs virtual machines on an IaaS infrastructure. He has full access to this VM. There should be no difference between running a VM inside a provider's IaaS infrastructure and a physical machine in the customer's own data center. The cloud customer implies the role of the *VM administrator*, also including the responsibility of software update and patch management of software within the VM. Furthermore, the cloud customer is responsible for the user and access management. A CSP can support the customer e.g., generating SSH key pairs, but deployment and safe storage is up to the cloud customer.

PaaS roles

In an PaaS scenario five roles can be identified: *cloud provider, service provider, cloud customer, service developer* and *user / user groups*. The definition of a *cloud provider* differs from the one of an IaaS provider. In PaaS, a cloud customer does not get full access to a virtual machine, he just gets access to a platform for running web applications. Therefore, a PaaS provider has additional responsibilities (as compared to an IaaS provider), such as installation and maintenance of the software stack inside a virtual machine e.g., OS, web server, database server, application server. Typically, an API for application development is provided by the PaaS provider. The *service provider* uses the cloud provider's platform to offer his service, typically as web applications, developed on the provided API. The service provider is a customer of the cloud provider. In a PaaS scenario the *cloud customer* is understood as a contractional partner with the service provider. He rents licenses for using the provided application from the service provider. If a cloud customer develops software on the cloud platform, he also inherits the role of the service provider. The *service developer* is typically employed with the service provider. He develops and maintains the offered (web) applications. Therefore, he needs access to the deployed (web) application to fulfil his role. The *user* or *user groups* represent the end customer of the offered (web) application, who actually work with it.

SaaS roles

In the SaaS scenario the same five roles exist, as in the PaaS scenario: *Cloud provider, service provider, cloud customer, service developer* and *user / user groups*, although their definition differs. The *cloud provider* supplies the platform for running services in the cloud. He also acts as a *service provider*, by offering (self-developed) services. The service provider is in a SaaS scenario a sub-role of the cloud provider. The cloud customer is a contractional partner of the service provider, renting software us-

age licenses. The *service developer* is typically employed with the service provider, developing and maintaining the offered (web) applications. The *user* or *user groups* represent the end customer of the offered (web) application, who actually work with it.

Independent of the underlying service model, the *Cloud Service Assembler* knows the cloud architecture and how to compose cloud services. He can be employed by the cloud provider or a third party utilising different cloud service providers.

2.7 Related Technologies to Cloud Computing

Cloud computing is related to the following technologies by sharing certain aspects: Virtualization, Client/Server model, Grid Computing, Utility Computing and Distributed Computing.

Virtualization is an enabling core technology for cloud computing. It allows the abstraction of physical hardware to provide virtual resources. A virtualized host is known as a Virtual Machine. Thus, it is possible to aggregate resources of one or multiple physical machines, such as CPU power, memory, network connectivity, disk storage, etc., and assign them dynamically as virtual resources to applications or services on-demand. Therefore, virtualization can be defined as a foundation technology of cloud computing [49].

The **Client-Server Model** is a **Distributed Computing** paradigm. It consists of a provider of a certain resource or a service (server) and somebody that requests this special service (client). Commonly a network is used for communication between servers and clients, but both can exist on a single host as well. "One or more Clients and one or more Servers, along with the underlying operating system and inter-process communication systems, form a composite system allowing distributed computation, analysis, and presentation" [50]. Cloud computing is building heavily on the client-

server model, since offered provider resources are consumed by a client.

Utility Computing represents a business model, in which computing resources are bundled as metered services and provided on-demand. Customers get charged based on their usage, rather than on fixed, rigid contracts. Cloud computing can be seen as one form of utility computing, since it provides dynamically packaged (virtual) resources and adopts the utility-based billing and accounting scheme.

Grid Computing delivers storage and computation capability over a network. To achieve this it "enables resource sharing and coordinated problem solving in dynamic, multi-institutional virtual organisations [51]. Typically grids are built by a number of corporations, professional groups or university consortia to serve a specific, specialised task, such as the Search for Extraterrestrial Intelligence (SETI@HOME) project. In SETI people are sharing unused processor cycles of their PCs to search for signs from outer space. While cloud computing shares a lot of Grid Computing's attributes (delivery of abstracted resources and services) it differs from it in security, the programming model, the underlying business, computation and data model, as well as applications and abstractions [32].

Cloud computing correlates with the addressed and other technologies of the distributed computing paradigm, such as Cluster- and Super Computing. But in contrast to cloud computing, Cluster- and Super computing are more application oriented, not intended to be shared in an on-demand fashion. Nevertheless, Cluster- or High Performance Computing can also be implemented in a service oriented delivery model (like cloud computing) as described in "ViteraaS: Virtual Cluster as a Service" [52].

2.8 Commercial Cloud Landscape

Due to the introduced cloud computing attributes and advantages, several business models have rapidly evolved to harness cloud computing through offering computing

infrastructure, programming and software application platforms or storage as a service. Due to the large amount of existing vendors in cloud computing it is hard to give a comprehensive vendor overview. However, Peter Laird published the first list of cloud vendors in 2008, which got updated in 2009 [53]. This was used as a basis for this research to give an overview of available cloud vendors, depicted in Figure 2.9. It shows a (non-comprehensive) taxonomy of cloud vendors. Another cloud taxonomy is provided by opencrowd under [54]. A survey by SearchCloudComputing.com [55] identified the following Top 10 cloud computing service provider in 2011. The ranking is based on "customer traction, solid technical innovation and management track record":

1. Amazon Web Service [23] - IaaS and Storage as a Service provider
2. Verizon/Terremark [56] - IaaS provider for telecommunications hardware
3. IBM [57] - provider of "Smart Business Test and Development Cloud", a non-public IaaS offer for enterprises
4. Salesforce.com [47] - a SaaS provider, which expands into PaaS services in 2011
5. CSC [58] - an IT integrator offering the private cloud service BizCloud
6. Rackspace - IaaS provider, publishes the free cloud software OpenStack [59]
7. Google App Engine [22] - offers PaaS services
8. BlueLock [60] - an IaaS provider offering VMware vCloud Express
9. Microsoft [45] - a SAaaS and PaaS provider (Windows Azure)
10. Joyent [61] - formed a partnership with Dell to sell preconfigured cloud infrastructure packages

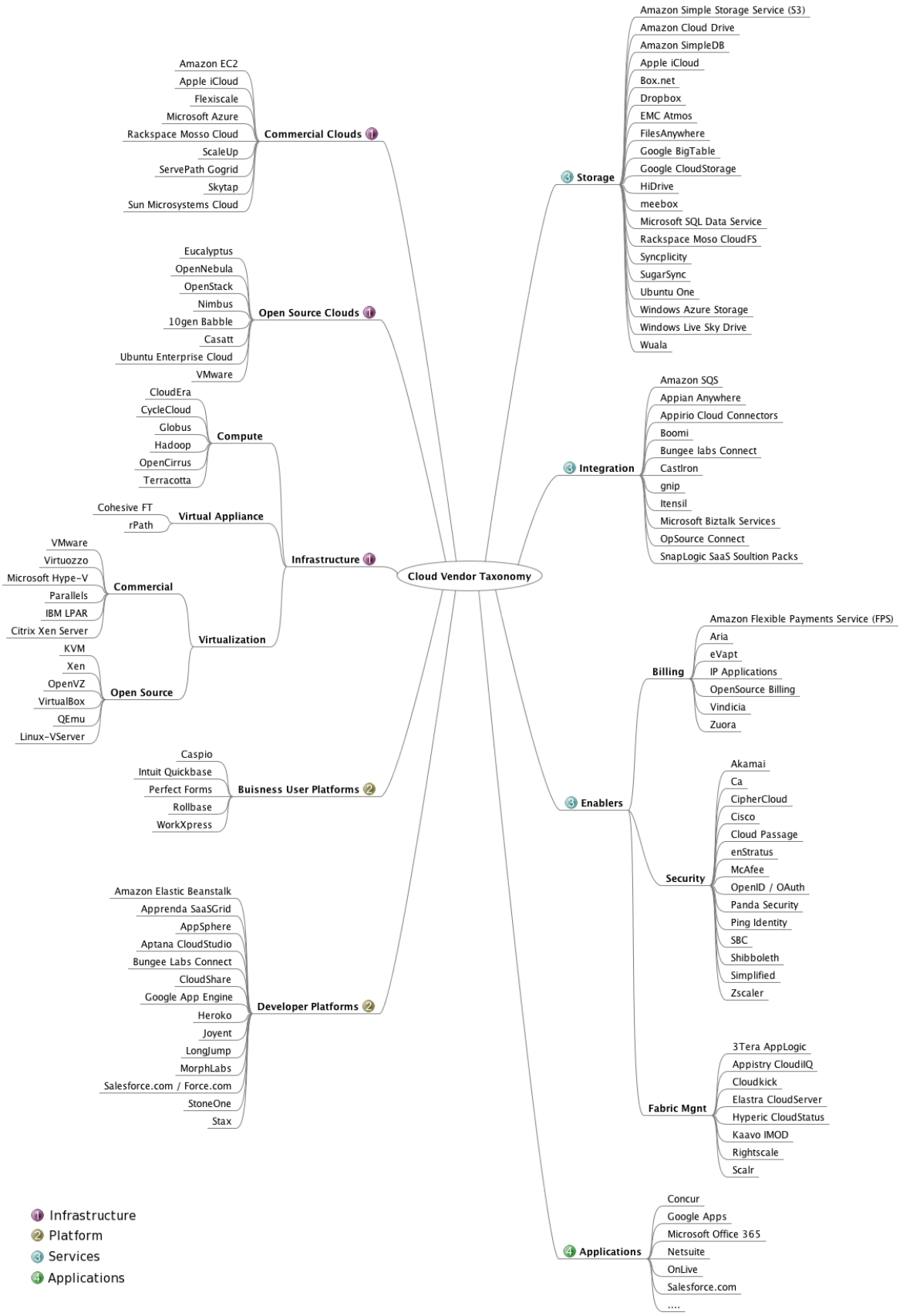


Figure 2.9: Cloud vendor taxonomy (4th quarter 2011)

One finding of the ranking is, that most of the Top 10 companies are "established cloud providers", being in the market for multiple years. Amazon was the first company offering IaaS services on a large scale since 2006, three years before the term "Cloud" was becoming a hype. Also IBM, Salesforce, Rackspace and Google participated in the cloud market from the very beginning. This also proves the evolution of cloud computing, from being a hype to a product developing, enterprise ready technology. Surprisingly, Dropbox Inc. is missing in this ranking, although they are providing the most widely know Storage as a Service product (Dropbox [18]) to end users.
Furthermore, a typical business development can be identified: as soon as an emerging technology transfers from just being a hype to a possible valuable business model, other established companies in this market are adding it to their portfolio as well. The ranking reflects this with companies like Microsoft, Verizon or Dell, which are established enterprises of the information technology industry now entering the cloud computing market as well.

2.9 Advantages and Disadvantages of Cloud Computing

Advantages of cloud computing exist for both involved parties: cloud customer and cloud service provider. For the cloud customer, e.g. small- and medium-sized enterprises, cloud computing enables them to avoid the over-provisioning of IT infrastructure and training personnel. Thus, SMEs can take advantage of using a cloud when the IT capacity needs to be modified on the fly. Typically, more resources are needed for services that are available only for a certain period. For example, AF83, a company specialising in social networking and live web solutions, used Amazon IT infrastructure to deliver a live concert via the Web and mobile devices [62]. The con-

cert attracted 7,000 simultaneous users. By using cloud computing, AF83 avoided purchasing new hardware for this special event, and delivered a successful service in a short amount of time. For companies with large IT infrastructure, such as Amazon or Google, becoming a cloud provider allows them to offer their resources to SMEs based on pay-as- you-go and subscription models, respectively. Because not all services require full resources at the same time for a long periods of time, these companies can still use and lease their existing infrastructure with a relatively small cost. Hence, they can reduce the total cost of ownership (TCO) and increase their own hardware utilisation [33]. The following advantages of cloud computing can be identified [49]:

No up-front investment. Cloud computing uses a pay-per-use service model. Therefore, cloud users can save large investments into data center infrastructure or software by renting it on-demand from a cloud service provider.

Lower operating and maintenance costs. Due to cloud computing's on-demand self service attribute rented cloud resources can be decreased or completely released in case of a low demand. This reduces operational costs. Typical costs, which can be lowered are hardware maintenance and staff training costs.

High scalability. Cloud resources can be pooled up to huge instances to satisfy peak load situations. Cloud user can declare thresholds, such as new connections per second to a certain cloud service, to define when a service should scale up or down. No provider interaction is necessary. This is also referred as "surge computing" [33].

Easy Access. Cloud services are easy accessible commonly either through a web based interfaces, such as a web page or an application programming interface. This can be a traditional PC or laptop as well as a tablet PC, a mobile phone or even just a browser plugin, like Amazon S3 Organizer [63].

Reduced business risk. In case of hardware failures, misconfiguration of services or missing maintenance (backup management), the business risk gets outsourced to the CSP who often has better experience and is better equipped to manage these risks.

Besides these benefits, there can also be disadvantages and risks identified in cloud computing:

Higher operating costs Cloud's access on-demand provides low up-front investment for customers to get an IT infrastructure. But, if multiple cloud resources are used permanently, cloud renting costs could quickly exceed traditional IT outsourcing models. A typical webserver scenario with 1 medium Amazon EC2 instances, used 24/7 during a whole month with 40GB of storage, 10% backup space, 10GB incoming and 50 GB outgoing network traffic results in estimated costs of $148.13 [24]. This compares to total costs of 40 EUR with a traditional root server hosting offer [64].

Network dependency By the nature of cloud computing, customers rely heavily upon the network connection, more precisely the Internet connection in a Public Cloud scenario. In case of connection problems or loss cloud resources are not available anymore. Offline usage is not possible.

Vendor dependency and lock-in In case of a security or availability incident, cloud customers depend highly on the cloud vendor to get information about what happened and when a service will be available again. In particular, "small" cloud customers are affected, as the CSP will unlikely consider their interest in a reasonable time frame. Because of missing cloud interoperability standards there is a particular risk of vendor lock-in. In SaaS, it is not so different to move from one solution to another, which is quite comparable to a traditional software application switch. However, the data format needs to be compatible. In a PaaS scenario there is a high risk of vendor lock-in especially through the given programming languages and interfaces. This is even increased if a proprietary language is used by the CSP. In an IaaS scenario lock-in can be less severe due to the use of (standardized) virtualization software. Anyhow, there is a risk due to incompatible hypervisor formats. Furthermore, VM images are typically very big (multiple Gigabyte), resulting in high bandwidth transfer costs when

moving them from one CSP to another. A research analysts from Gartner even sees lock-in and missing standards even surpass security as the biggest objection to cloud computing [65].

Data security Data is an important business asset. Since a proper, CSP independent security model is not developed yet, there is a loss of control over data in cloud computing. This is mainly because of: unknown physical location of hardware and software (especially in Public Clouds), absence of cloud security standards, lack of compliance standards, such as HIPAA [66], SOX [67], PCI [68], SAS 70 [69] and a risk of data loss due to improper backups or system failures in the virtualized environment. Chapter 3 elaborates on this further.

2.10 Summary

In this chapter the evolution and key attributes of cloud computing were introduced. A definition of cloud computing was given, and common cloud computing service and deployment models were discussed. Basic service components of cloud infrastructures were shown and studies were presented, which prove, that cloud computing technology is important to industry and not just a hype. Related technologies were introduced and how cloud computing is distinguished from them. A cloud vendor taxonomy informed about the commercial cloud landscape and advantages as well as disadvantages of cloud computing were discussed.

3 Cloud Infrastructure Audit

"Security is a process, not a product"

(Bruce Schneier, Security technologist)

After setting the context and showing related work, this Chapter presents the first main phase of the research: the analysis of cloud specific security challenges and how they can be approached with cloud security audits. An audit criteria test catalogue gets presented at the end of this Chapter, forming the first novel contribution of this research.

3.1 Introduction

Pushed by cloud vendors promising "infinite scalability and resources" combined with on-demand access from everywhere, cloud user quickly forget that there is still a real IT infrastructure behind a cloud, where the architectural complexity is actually increased compared to traditional data centers. The chapter starts with discussing, why there is a high research demand on cloud computing security and underpinning it by introducing disclosed cloud computing security incidents at a commercial cloud provider. A study on the security issues relevant to cloud computing is presented. Subsequently, security audits are introduced and it is elaborated, how they can be utilised to mitigate the identified cloud security challenges. As a result, a cloud audit test criteria catalogue is developed.

Summary of Research

This part of the research analyses cloud specific security issues and shows how security audits can be adapted to mitigate the identified challenges.

Parts of this research phase have been published in the following papers and presented to the research community at the corresponding conferences:

- Designing Cloud Services Adhering to Government Privacy Laws, *3rd IEEE International Symposium on Trust, Security and Privacy for Emerging Applications 2010* [70]
- Security issues in IT outsourcing through cloud computing, journal *Practical Commercial Information Technology* [71] - Winner of annual Best Paper prize
- Understanding Cloud Audits, book chapter in *Privacy and Security for cloud Computing 2013* [72]

3.2 Security Concerns in Cloud Computing

Beneath the advantages of cloud computing, enterprise analysts and research have identified cloud specific security problems as the major research area in cloud computing [73, 74, 75, 76]. In a survey amongst 235 CIOs and other IT executives at leading U.S. companies with annual sales of more than $500 million USD [8], security concerns were the major issue which hinders a broad industry acceptance of actually utilising cloud technologies, as illustrated by Figure 3.1. The fact, that also for private cloud solutions (where an enterprise uses cloud technology only within their own IT infrastructure) security concerns are named as the major obstacle suggests, that there is a general lack of trust in cloud computing security. Since security is still a considerable challenge for classic IT environments, it is even more for cloud environments due to its unique characteristics, such as: seamless scalability, shared resources, multi-tenancy, access from everywhere, on-demand availability and third party hosting. Whereas a security incident in a traditional online system can mainly lead to one specific compromised system, in cloud computing this risk is much more severe, since a compromised cloud instance can result in the compromisation of all IT services of a specific company run within the cloud. Follow-up attacks are expected, which could also lead to an intrusion into further areas of a companies IT, not necessarily run within a cloud. Although existing industry recommendations (ITIL), standards (ISO 20000, ISO 27001, CobiT) and laws (e.g., Germany's Federal Data Protection Act) provide well established security and privacy rule sets for data center providers, research has shown that additional regulations have to be defined for cloud environments [73, 70]. The following examples of cloud security incidents assist in illustrating the need for better cloud computing security:

- Hackers stole credentials of Salesforce.com customers via phishing (2007) [77]

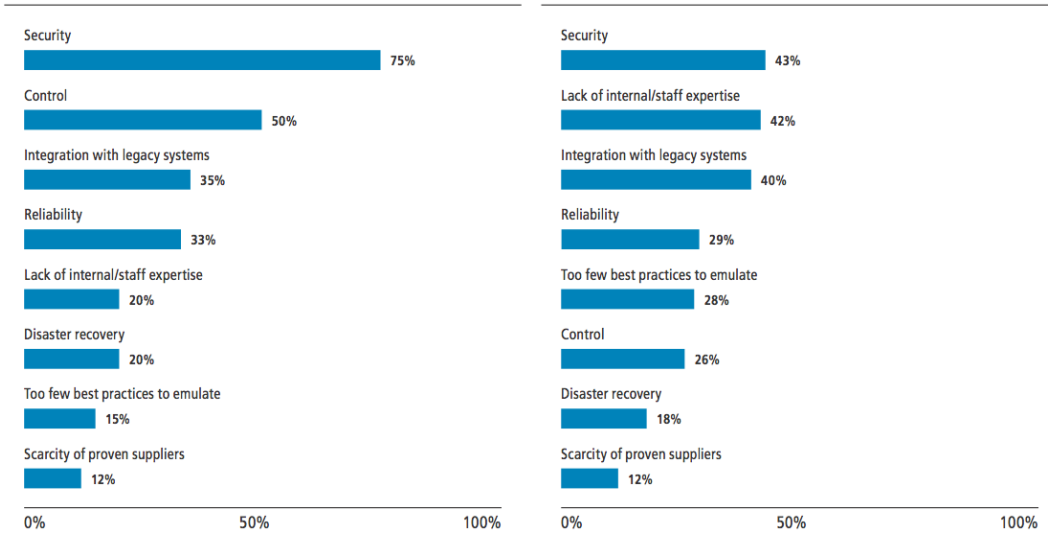


Figure 3.1: Top concerns on cloud computing at organisations [8]

- T-Mobile customers lost data due to "Sidekick disaster" of Microsoft cloud (2009) [78]
- Botnet incident at Amazon EC2 infected customer's computers and compromised their privacy (2009) [79]
- Amazon customer services were unavailable for multiple days and data was lost due to a logical flaw in the cloud storage design (2011) [80, 81]
- Hotmail accounts were hacked due to technical flaws in Microsoft software (2012) [82]

To show, that traditional IT security best practices can not just be applied to cloud computing environments, the Amazon security incident serves as an example: After an infrastructure outage in April 2011, Amazon's Compute Cloud EC2 was not available causing popular services like Reddit to be unable to serve its customers [80]. While such an outage also violates EC2's quality of service level agreements of 99.95% availability: ($\hat{=}$ 1,825 outage days/year), Amazon's support handling caused a strong loss of trust in it as a cloud provider due to the following:

(1) Amazon data centers are divided into several availability zones to distribute the impact of (hardware) failures. For resilience reasons users distribute their data over

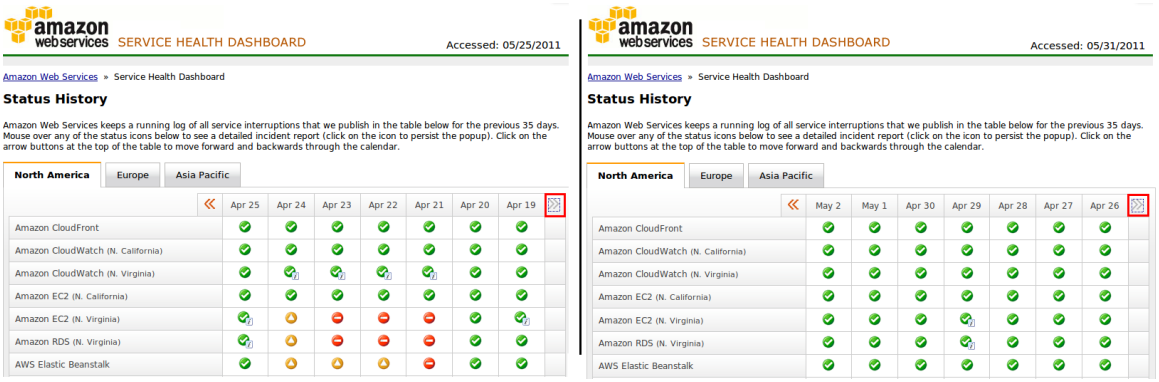


Figure 3.2: Amazon Health Dashboard accessed at 05/25/2011 and 05/31/2011

different availability zones. As a result of the outage EC2 customers permanently lost data, although services were hosted on different EC2 availability zones [81]. A company offering webservice usage monitoring lost 11 hours of historical data.

(2) During the crash an EC2 customer running a monitoring service of cardiac patients tried to reach Amazon's support unsuccessfully. Neither information about the expected downtime nor moving the unreachable instances to a different EC2 data center was offered.

(3) Since hardware sovereignty is given away in cloud computing security, health and monitoring information is critical to cloud users to build there services in an appropriate way regardless which cloud model (public, hybrid, private cloud) is used. This is already known from traditional IT outsourcing and providers try to establish trust to customers by proving their compliance to IT security standards like ISO27001 [83] or ISO9001 [84]. Amazon AWS so far seams to follow a contrary approach: although AWS provides status information about the cloud infrastructure at the Amazon Service Health Dashboard [85] users can only see a service' history over the last five weeks. Amazons problems from April 2011 were not visible anymore to users by the end of May 2011, as depicted in Figure 3.2.

Maintaining consistent security across boundaries is complex and challenging for information security professionals [1]. The Cloud Security Alliance defined a cloud model

Layer	**Service model**		
	Software as a Service	**Platform as a Service**	**Infrastructure as a Service**
Facility	✓	✓	✓
Network	✓	✓	✓
Hardware	✓	✓	✓
Operating System	✓	✓	?
Middleware	✓	?	-
Application	✓	-	-
User	-	-	-

Table 3.1: What the cloud provider controls [14]

consisting of seven layers: facility, network, hardware, operating system, middleware technology, application and user. Table 3.1 shows in which cloud service model which cloud architectural layer is controlled by the cloud provider [14].

3.3 Related Work

One of the first professional groups, which published about cloud computing security, is the Cloud Computing Security Alliance (CSA). CSA is an organisation with around 120 corporate members and has a broad remit to address all aspects of cloud security, including compliance, global security related legislation and regulation, identity management, and the challenge of monitoring and auditing security across a cloud based IT supply chain [86]. Its main work is the "Security Guidance for Critical Areas of Focus in Cloud Computing" [86], which identifies several key security domains and topics, such as compliance and audit, incident report and encryption. It provides several recommendations on each domain. The guide was first released in April 2009, the current version 3 was released in November 2011. It is known as a de facto standard document when it comes to security guidelines for cloud computing. In addition, CSA released the guideline: "Top Threats to Cloud Computing V1.0" [26] in 2010. It iden-

tifies the most urgent threats, which need to be addressed in cloud computing. The guide was updated in 2013 as "The Notorious Nine - Cloud Computing Top Threats in 2013" [87]. Although many major cloud security issues were identified by CSA, the work of this thesis partially extends them by the issues "Data life cycle in case of provider termination" and "Missing monitoring of cloud scalability", presented in *Section 3.4 - Cloud Computing Security Issues*.

The European Network and Information Security Agency (ENISA) publishes the guide "Cloud Computing Security Risk Assessment" [74]. It presents a survey, which identifies security for cloud computing and especially the need for a trustable, higher assurance cloud architecture as the major cloud related research area to be addressed. Vulnerabilities, technical and legal risks as well as a demo scenario of an SME moving into cloud computing gets presented. A high level questionnaire to evaluate the risk of moving services into a cloud environment is provided. Although this work is a step into the right direction, it still stays to high level when it comes to recommendations for a secure utilisation of cloud technology. It is becoming clear, that there is an urgent need for a detailed security check list or catalogue for the cloud consumer, which can be given to the provider to evaluate if necessary security features are in place to protect customers data and address cloud specific security issues.

The German Federal Office for Information Security (Bundesamt für Sicherheit in der Informationstechnik (BSI)) published the white paper "Security recommendations for cloud computing provider" [88] in 2011. It provides detailed recommendations, how cloud providers should secure their infrastructure and which questions should be asked a provider to achieve a high level of security and governance to data protection laws. The white paper was updated by security experts from industry and academia and released as a technical guide in 2012 [89]. Although it is very detailed, some cloud specific security issues are missing, including almost all possible attacks on the cloud management system, such as scalability attacks, and resultant billing and accounting

issues. Again, this identifies the urgent need for a technical detailed analysis on cloud specific security issues, resulting in a cloud audit criteria catalogue for cloud consumer and provider. Thus, contact with the BSI was established and results of the research were submitted to the BSI to contribute in their effort. However, it remains unclear if the BSI utilised the results within the guide. All the presented guides, provide recommendations for cloud users especially on the management level. However, a specific architectural proposal, how the suggestions made can be implemented is missing. The research work proposed in this thesis intends to fill this gap.

Cloud Security Research Papers

A rather high level, but comprehensive perspective on the whole topic of cloud computing security is given by Mather et al in the book "Cloud Security and Privacy" [1]. It provides a very good entry into the topic especially by laying out the necessary groundwork. Due to the wide area the book covers it can't go very deep into the necessary topics important for this research.

The most comprehensive survey about current literature addressing cloud security issues is given by Vaquero et al. in [75]. It categorises the most widely accepted cloud security issues into three different domains of the Infrastructure as a Service model: machine virtualization, network virtualization and the physical domain. It also proposes prevention frameworks on several architectural levels to address the identified issues. Pearson [90] proposes several software design guidelines for delivering cloud services taking privacy into account, such as using a privacy impact assessment, allowing user choice and providing feedback. While Chen et al. state in [76] that many IaaS-related cloud security problems are problems of traditional computing solved by presented technology frameworks it also demands an architecture that enables "mutual trust" for cloud user and cloud provider. Both papers confirm and complement the

cloud specific security issues identified by this research. Furthermore, they identified a demand for a two-way trust enabling architecture for cloud infrastructures and the ability of "choosable security primitives with well considered defaults" [76]. The SAaaS architecture, presented in Chapter 4, will provide this mutual trust. SAaaS' Cloud Audit Policy Language (see Chapter 5) enables the user to define its own security levels and to choose from a spectrum of security subsystems as demanded by [76].

Amazon's cloud platform Elastic Compute Cloud (EC2) allows users to create and share virtual machine images. Balduzzi et al. [91] analyzed the security risks of running third party images. The work gave a good insight about the current risk, which comes from pre-configured cloud appliances. After the investigation of over 5000 Amazon Machine Images (AMIs), they found that 98% of Windows AMIs and 58% of Linux AMIs contain software with critical vulnerabilities [91]. Furthermore, two VM images were infected with malware, two were configured to write logs to an external machine, 21.8% contained leftover credentials that would allow a third party to remotely log into a machine [91]. Their approach is to start AMIs on EC2 and then scan them for security problems and privacy risks. However, their system is not intended to be used as an auditing service. They execute a predefined set of security and privacy checks and provide no way of customizing policies, which will be supported by the work proposed in this research.

Other Cloud Audit Projects

To address the lack of existing standards regarding cloud specific security problems, a number of open audit projects have been created. A working group of the Cloud Security Alliance is the umbrella project Governance, Risk Management and Compliance (CSR) [92]. It includes the sub projects "Cloud Audit A6", "Cloud Controls Matrix (CCM)", "Consensus Assessments Initiative Questionnaire (CAIQ)" and "Cloud Trust

Protocol (CTP)". The Cloud Audit A6 - Automated Audit, Assertion, Assessment and Assurance API [93] has the goal to provide a common interface and namespace for cloud computing providers to automate the audit, assertion, assessment, and assurance of their cloud environments. The Cloud Controls Matrix "is specifically designed to provide fundamental security principles to guide cloud vendors and to assist prospective cloud customers in assessing the overall security risk of a cloud provider." [92]. A questionnaire about which security controls exist in IaaS, PaaS and SaaS offers is provided by the Consensus Assessments Initiative Questionnaire. Finally, the CTP is an initiative to create a machine and human readable protocol which provides Transparency as a Service for cloud users about compliance of cloud provider [94].

The "EuroCloud Star Audit" [95] is a German certification for SaaS providers. It aims to establish a high level of security and transparency for users and providers alike. The audit starts with the provider's general profile, carries on with contract and compliance including data privacy protection, general security, operation and infrastructure, operation processes and goes as far as application and implementation.

Wang et al. present in [96] a system to audit integrity and security of public data cloud storage. Their solution allows a third party auditor to be able to efficiently audit the cloud data storage without demanding the local copy of data, and introduce no additional on-line burden to the cloud user. Therefore they combine the public key based homomorphic authenticator with random masking to achieve a privacy-preserving public cloud data auditing system.

A "Dynamic Audit Services for Outsourced Storages in Clouds" is presented by Zhu et al in [97]. The approach uses fragment structures, random sampling and index-hash tables, supporting provable updates to outsourced data and timely anomaly detection.

Massonett et al. discuss in [98] the problem for IT security audits if federated cloud infrastructures are spanned across different countries. They introduce a existing federated cloud monitoring infrastructure to monitor in which country data is actually

saved without compromising cloud isolation. In the presented approach, collaboration is required between the cloud infrastructure provider and the user of the cloud, the service provider. The proposed architecture is validated by an e-Government case study with legal data location constraints.

Tancock introduces in [99] a privacy impact assessment decision support tool that can be integrated within a cloud computing environment. The authors show that privacy weaknesses impact legal compliance, data security and user trust in cloud environments. The presented system is a systematic process for evaluating the possible future effects that a particular activity or proposal may have on an individual's privacy. With the system presented, risk analysis of moving a service to a cloud environment can be enhanced. A distributed monitoring facility can deliver the input to detect multiple Cloud-specific security issues. Li et al. present in [100] a method how cloud storage services can benefit from a trusted third party audit (TPA). They introduce issues and solutions for the application of a TPA, such as protection for data integrity, support of dynamic data, access control batch audits and minimised audit costs.

Although the introduced Cloud Control Matrix by CSA and guidelines by ENISA and BSI are a first step into the right direction, they lack a necessary technical depth when it comes to security guidelines and corresponding checks. Currently, most audit guidelines in cloud computing are either very high level or are focussing on the specific model of Software as a Service clouds. However, there has been little effort made on auditing of IaaS clouds. Detailed security criteria in form of technical and organizational IaaS audit checks are necessary, which can be used by the cloud customers to evaluate the security status of a cloud. The presented EuroCloud Star Audit would deliver this technical depth, however it is only targeted for Software as a Service clouds. Such a detailed audit criteria list is missing for IaaS clouds. Thus, as a first part of this research an IaaS audit criteria catalogue needs to be developed, which delivers this missing technical depth.

The introduced related research work is introducing valuable security concepts for clouds. However, they are concentrating on a very specific types of IaaS clouds, such as Storage as a Service clouds ([97]). Others ([100]) are requiring the usage of additional or special hardware (TPM) or are assuming a inter-provider collaboration ([98]), which does not exist, due to to lack of cloud standards. Therefore, a cloud audit system, which is targeted for general IaaS clouds and does not introduce the need of special hardware is needed. It should be simple and based on standardised hardware and software. It needs to be able to deliver audit information, independent on the underlying cloud provider's infrastructure.

3.4 Cloud Computing Security Issues

A detailed analysis of the actual security impact is not easy to find, because very often security problems are declared as cloud security problems, although they already exist in traditional IT-outsourcing scenarios and are merely exacerbated in cloud environments. Figure 3.3 shows an updated version of Figure 2.8 (Section 2.5 - Cloud Computing Architecture) highlighting typical "hot spots" of traditional and cloud specific security risks in an IaaS cloud environment.

The German Federal Office for Information Security publishes the IT baseline protection catalogues [101], enabling enterprises to achieve an appropriate security level for all types of information. The catalogues were extended by a special module covering virtualization in 2010. As part of the research presented in this report, a comprehensive study on all available IT baseline protection catalogues as well as current scientific literature available ([73, 26, 74, 75, 76]) was published. A comparison and origin classification of security issues from classic IT-Housing, IT-Outsourcing and cloud computing was undertaken. New cloud specific security issues, as well as already known security issues which are amplified by the usage of cloud computing were

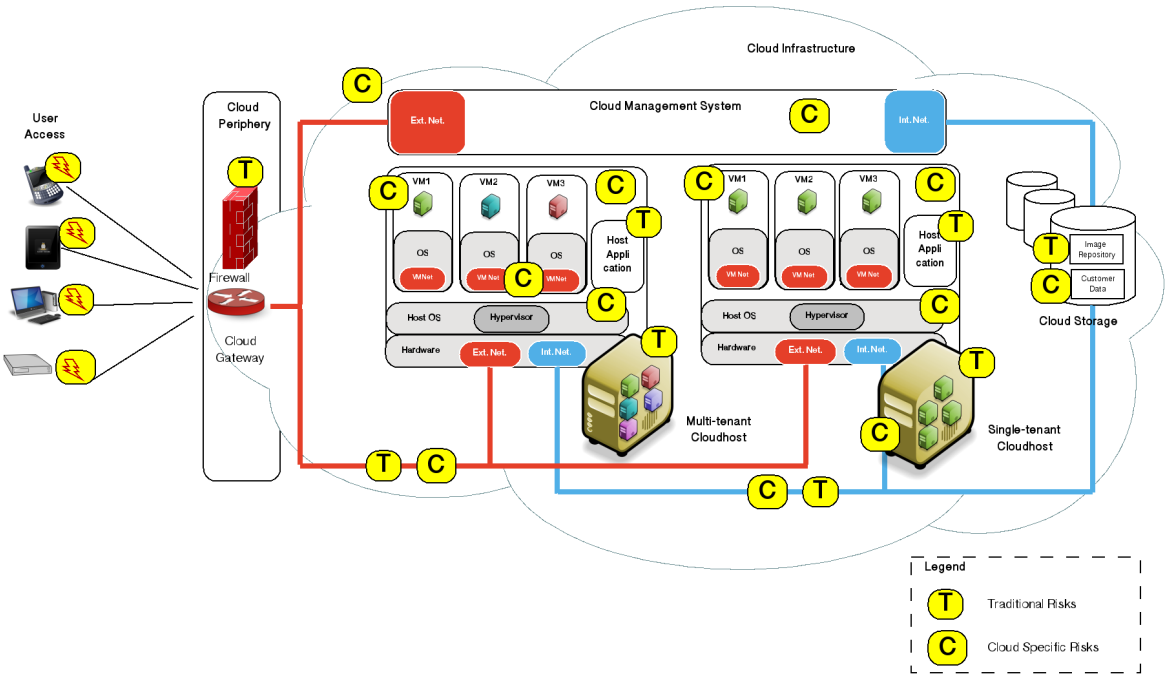


Figure 3.3: Security Risks in a typical IaaS cloud environment

identified. Recommended security best-practice of security issues which are amplified by cloud computing are evaluated due to their applicability. This section will list cloud security problems and compare them to similar problems already known from traditional IT outsourcing. Furthermore, each identified issue is categorised by the affected core principles of information security, such as availability, confidentiality, integrity [102].

3.4.1 Cloud Computing vs. Classic IT Outsourcing

In the definition of traditional "IT outsourcing" the two most common models existing on the market are combined: IT housing and outsourcing. In IT housing, a customer provides its own hardware, e.g. server, and just runs them in a data center of a service provider. He is only providing the necessary infrastructure components, like network components, cooling or power. Administration of the hardware stays with the customer. In outsourcing, a customer rents the complete infrastructure from a service

provider, including any hardware and software. Administration is undertaken by the service provider. In traditional IT outsourcing administration or business processes get, partly or fully externalised to a third party service provider. A customer is renting a certain infrastructure and using it exclusively, which is called a *single tenant model*. An extension of infrastructure or service requires a prior communication with the service provider. Long contract durations are characteristic for traditional IT outsourcing. In cloud computing, a customer is also renting a certain infrastructure but shares them most of the time with other customers. This is identified as the *multi-tenant model*. Scalability of the rented service is simple, automatable and adaptable without prior interaction of the cloud provider. Cloud computing contracts are flexible in duration and can vary from just a couple of minutes to years. A number of research papers [103, 76, 74], etc. identified cloud security and privacy problems. They all have in common the following differentiation of identified cloud-specific security problems:

- Amplified Cloud Security Problems (amplified CSP): problems, already known from traditional, distributed IT-environments, but amplified through cloud computing attributes
- Specific cloud Security Problems (specific CSP): security problems which arise due to cloud computing's special characteristics.

In a comprehensive study on academic literature, best practices and standard recommendations on cloud computing security were analysed to identify cloud security issues. An important source for security guidelines in Germany are the IT Security Baseline Catalogues [101] published by the German Federal Office for Information Security. Thus, they were analysed to identify which existing security recommendations already cover cloud security issues, and how they need to be updated or completed. The following list of cloud specific security issues is the result of this investigation.

3.4.2 Amplified Cloud Security Problems

Amplified cloud security problems (amplified CSP) are mainly originated by underlying technologies upon which cloud computing is substantially built, such as, virtualization technology, web applications and multi-tenant software architectures. Furthermore, problems originating in well known and commonly established security best-practices, which are difficult or impossible to implement in a cloud computing environment are also classified as amplified CSP. The following amplified CSP have been identified:

A1: Misuse of administrator rights / malicious insiders

Misuse of administrator rights is a severe problem already known in traditional IT. In a recent survey [104] among 300 IT professionals 26% admitted, that at least one staff member has abused a privileged login to access information. In cloud computing this threat is amplified. Virtual machines are mostly provided as managed root servers. The cloud provider is responsible for the underlying host system and has access to the VMs running on the host through the hypervisor. A misuse through malicious insiders is possible and hard to detect due to a general lack of transparency into provider process and procedure. This affects the following core principles of information security: *confidentiality, authenticity, authorisation, integrity, data protection, accountability and non-repudiation.*

A2. Missing transparency of applied security measures

In traditional IT outsourcing, this risk is mitigated by well defined regulation: The customer (IT housing) or the provider (IT outsourcing) is responsible for the application of security measures. They must be communicated to the customer. Providers can prove their compliance to baseline security measures with ISO 27001 or PCI DSS

certificates. In cloud computing there is a lack in transparency regarding applied provider security measures and processes exists. The underlying hardware infrastructure gets masqueraded to protect it from attacks. Cloud customers currently need to trust the provider that they are compliant to current security standards. Amazon Web Services announced in December 2010 that the AWS data center, infrastructure and services are compliant to ISO 27001 and PCI DSS Level 1 [105]. However, to date no agreed standard criteria for running a secure cloud infrastructure exist. This affects the following core principles of information security: *integrity, availability and data protection.*

A3. Missing transparency with security incidents

Since computing systems are completely owned by the customer in IT housing, they are responsible for securing all evidence in case of a security incident. In IT outsourcing this responsibility is transferred to the service provider which employs skilled personnel, e.g. an own Computer Emergency Response Team (CERT). In cloud computing, customer and provider need to work together to collect all information of a security incident. Problems with hardware must be mapped to the different customer cloud resources to react on incidents and initiate correct problem management. But a standardized procedure is currently missing. Current cloud offers available in the market do not offer a transparent process for its customers on how security incidences are detected, which efforts are taken by the provider to mitigate it and how the provider supports its customer during the investigation phase. This is an increased risk in cloud computing. This affects the following core principles of information security: *data protection, integrity, availability and non-repudiation.*

A4. Shared technology issues

This threat includes the problem of sharing physical resources with multiple customers

Year	Name	Description
2007	vmftp [106]	Directory traversal vulnerability in VMware tools
2009	XEN Ownage Trilogy [107]	Exploiting drivers using DMA
2009	VMware Cloudburst [108]	Abuse of a lack of access control in a VMware 3D graphics driver
2011	KVM Virtunoid [109]	Dangling pointers due to a bug in the hardware emulation layer
2012	VMware VMSA-2012-0009 [110]	a) Bug in the Backdoor API (for communication between VMware tools and host) channel between VM and host b) Bug in the SCSI device registration (no further details available, potentially bug in hardware emulation layer) c) Buffer overflow in floppy driver (no further details available, potentially bug in hardware emulation layer)
2012	VMDK Has Left The Building [111]	Design flaw in the VMware ESXi hard disk handling
2012	XEN Sysret [112]	Para virtualization API design flaw

Table 3.2: Overview of Hypervisor outbreaks [15]

as well the problem of misconfigured VMs that endangers other resources. In IT housing this threat only applies for misconfiguration of security parameters and is limited to one corresponding customer. In IT outsourcing the provider is fully responsible to configure running services securely. In cloud computing this is caused by the use of virtualization. Table 3.2 shows security incidents based on hypervisor outbreaks to illustrate the increased risk for cloud computing. The main problem is the lack of isolation, which for cloud computing can be categorised into:

- VM isolation: If one customer runs an improperly configured VM in the cloud this also endangers other VMs running on this specific host. An attacker could use a VM as an entry point to get access to the host machine through a hypervisor flaw to gain inappropriate levels of control or influence on the underlying platform. Exploits seem rare but have already been demonstrated by Kortchorski [108] and Rutkowska [113]. Although few successful attacks are published so far, increasing code complexity in hypervisor software amplifies this threat.
- Memory/Cache isolation: Often, the underlying components that make up this

infrastructure, e.g. GPUs or CPU caches were not designed to offer strong isolation properties for a multi-tenant architecture [26]. These resources need to be quickly allocated and de-allocated to fulfil a current demand. Well established measures for secure data wiping might not be applicable. So far no cloud provider discloses information on how shared resources get securely wiped before being reassigned to a different customer. Furthermore, by getting a default root access to a VM in current IaaS offering enlarges the attack vector of breaking through the isolation of shared resources. Certified Common Criteria compliant hypervisor software (minimum EAL 4) could mitigate this threat [88].

- I/O isolation: if there are problems with the virtual network (bridge software) traffic sniffing can be undertaken by an attacker.

Another security risk comes from the usage of pre-provided virtual machine images. The number of administrators of a traditional data centre is limited and they all are working under the same company security policy, while installing and maintaining machines. This can be completely different in a cloud infrastructure. Public marketplaces for exchanging cloud appliances such as, OpenNebula Marketplace [114], Amazon Web Services EC2 Management Console or the Amazon Web Services Marketplace [115] provide cloud customers with an easy and efficient way of finding the right virtual machine image. But they also allow users to be administrators of their virtual machines, or upload and share their custom made VM images with other users. Although cloud providers provide security guidelines [116] on how to prepare an image before releasing it to a marketplace, current research by Balduzzi [91], Bugiel [117] and Meer [118] shows, that marketplace images are highly insecure due to old software versions or "forgotten" or restorable security credentials, such as SSH private keys. Users, uploading appliances are usually more or less anonymous. There is no way to easily determine whether a custom appliance is legit or maliciously manipulated.

Images could contain rootkits, which are performing passive eavesdropping attacks such as traffic analysis, keylogging or transmission of user's data to external systems for industrial spying [91].
This affects the following core principles of information security: *integrity, availability, data protection, confidentiality, authentication and non-repudiation.*

A5. Data life cycle in case of provider switch or termination

This threat does not exist in IT housing since data and computing resources remain the property of the customer if he changes the housing provider. In IT outsourcing service level agreements control how data is transferred to a customer or how storage devices need to be securely wiped or disposed of. In cloud computing this threat is increased due to shared usage of resources. Customers need to define special rules for end of contract scenarios regulating how data gets exported from the cloud and how a provider has to securely erase customer's data [26]. This affects the following core principles of information security: *data protection and confidentiality.*

A6. Monitoring of Service Level Agreements

IT housing and IT outsourcing can easily log events per user. In a cloud several multi-tenant applications running in a virtualized environment need special tools to monitor Service Level Agreements. New tools for hypervisor, virtualized networking monitoring, etc. must be available. This affects the following core principles of information security: *availability and integrity.*

3.4.3 Specific Cloud Security Problems

Due to the prior given definition, specific cloud security problems (specific CSP) exist, when they originate or affect at least one of NIST's cloud characteristics.

B1. Unclear data location

In traditional IT outsourcing a customer always knows where and from whom its data gets stored and processed. Mostly, customers can physically visit a data center to inform themselves personally about the security measurements a provider has taken for data protection. Germany's data protection act §11 (1) states that where other bodies are commissioned to collect, process or use personal data, the responsibility for compliance within the provisions of this act and with other data protection provisions shall rest with the principal [119]. From the interpretation of this act, users must know the exact location of their data and their cloud providers' court of jurisdiction. An export or move of data is not possible without prior notification of the customer. In current cloud computing offerings customers do not have the possibility of knowing where the data gets stored or processed. Only a very rough decision about a cloud data centre's continental location can be made, e.g. AWS data center in Northern Ireland. Nevertheless, currently there is no way to prove if data is not outsourced by a cloud provider. A current court decision about legitimate access of US governmental agencies to data of US originated firms even of data centers located outside of US area of jurisdiction [120] strongly amplifies this risk. This affects the following core principles of information security: *data protection, confidentiality and availability.*

B2. Abuse and nefarious use of cloud resources

Characteristic for cloud computing is fast access to numerous virtual machines within a very short time frame. This attracts not only legally acting enterprises or organisations but also individuals and organisations with more malicious intent. Amazon's cloud was already used to host malware (e.g. trojans). Also the Zeus botnet (a phishing trojan that steals banking information) was known to be hosted on virtual machines within the Amazon cloud. Another possibility would be to aggregate many VMs and use them to DDoS a single target and thereby preventing others to use its

services. While this threat mainly addresses the cloud provider, the cloud customer can also be affected. As a result of the Zeus Botnet big parts of Amazon's IP address range was blacklisted on spam lists causing e-mails from "good" customers, running their mail server in Amazon, being rejected as well. This issue affects the following core principles of information security: *availability*.

B3. Missing Monitoring

A security incident within a cloud environment should get detected and eliminated by the cloud provider. If customer data is in danger, this should be communicated. To the best knowledge of the author, no cloud provider so far runs an information system that will inform the customer automatically. But especially in cases of personal data processing, e.g. credit card information, it could be important for a cloud customer to know if a security problem exists, so he can stop the service to guarantee data protection and integrity and minimise risk for its own systems. For a sustainable risk analysis of running a service in a cloud it is important to know:

- Which data protection measures exist to secure the cloud environment (anti virus protection, Intrusion Detection Systems (IDS), measures for Denial of Service (DoS) detection and prevention, patch and change management)
- History of service breakdowns
- Measurements taken for availability, backup, reliability and data recovery
- Installed software versions at cloud host systems and periphery
- Tracking of administrative access of cloud provider service personnel
- What information and support is available during a service breakdown or a security incident.

For monitoring security of large IT infrastructures a best practice approach is to run

an Intrusion Detection Systems with distributed sensors as input feeds. But this approach breaks down for cloud infrastructures, mainly, because of the complexity and frequently changing environment driven by the users. Traditional IDS setups are built around a single monolithic entity, which is not adaptive enough to do data collection and processing in an efficient and meaningful way [121]. This affects the following core principles of information security: *non-repudiation, availability, data protection and confidentiality.*

B4. Insecure Application Programming Interfaces

Cloud resources are mostly deployed, controlled, orchestrated and managed through specific cloud Application Programming Interfaces offered by the provider. The security and availability of general cloud services is dependent upon the security of these basic APIs. From authentication and access control to encryption and activity monitoring, these interfaces must be designed to protect against both accidental and malicious attempts to circumvent policy [26]. Since third party providers essentially build their services upon these APIs, e.g. load balancer service, a complex architectural layer gets inserted which needs to be subject to careful investigation. Standardized protocols and measurements for secure software development (Microsoft Secure Development Life cycle (SDL) or Software Assurance Maturity Model (SAMM) of the Open Web Application Security Projects (OWASP)) address this threat. This affects the following core principles of information security: *confidentiality, integrity, availability, non-repudiation, data protection and accountability.*

B5. Missing monitoring of cloud scalability

One reason for using a cloud infrastructure is to benefit from its scalability attributes. In this context it is most often used to deal with usage peeks, for example if a new version of software gets released and huge download requests are expected. Char-

acteristic to peeks is that they are mostly foreseeable and limited to a certain time frame. Therefore, cloud users design their cloud application to start new instances if a certain threshold is reached to provide service availability. This introduces two new challenges for cloud security:

B5.1) IaaS up-scaling - business driven Since a user's infrastructure can change rapidly (grow, shrink) in case of a peek scenario a monitoring system needs to be aware of the peek situation and the defined scalability thresholds.

B5.2) IaaS up-scaling - attack driven Most of the time scalability thresholds, like "maximum number new VMs to be created" get defined once. Mostly during the design phase for the first peek event. If the peek was managed well by the thresholds they just stay, like defined, although they might be not needed anymore (e.g., until the next major version release). This enables a new cloud specific attack: Financial damage due to nefarious abuse of cloud resources. An attacker can cause the creation of new cloud instances up to the scalability threshold by creating a huge number of allowed requests, which do not result in any successful business case but could be caused by e.g., distribution of malicious software.

This affects the following core principles of information security: *availability and accountability.*

B6. Missing interoperability of cloud provider

To minimise the potential damage of a provider downtime or in case of a provider change, interoperability between different cloud providers is very important. Current cloud offerings are not compatible with each other due to the usage of customised VM formats or proprietary APIs. A migration of cloud resources from one provider to another is not possible. This increases the risk of vendor and data lock-in. For

example, a customer of a Microsoft Azure database service can not use it with a service, developed and running on the Google App Engine cloud [88]. Standards are necessary to mitigate this risk. First developments are started with the following projects:

- Open Cloud Computing Interface
- Open Virtualization Format (OVF)
- Open Stack Cloud Software by Rackspace Hosting, NASA

Furthermore, a detailed strategy needs to be defined between provider and customer which regulates data formats, perpetuation of logic relations and total costs in case of a provider change [88]. This affects the following core principles of information security: *availability.*

B7. Increased complexity due to cloud characteristics

Cloud computing environments are far more complex than traditional data centers structures. Therefore, known best practices may not be enough to mitigate a security incident as shown in the Amazon example in Section 3.2. While having a backup server image might be enough in a traditional data center, this does not necessarily apply for cloud environments. Security and business continuity measurements best practices need to be re-evaluated for cloud infrastructures. This affects the following core principles of information security: *availability, integrity, confidentiality.*

Table 3.3 summarizes the presented cloud security problems (amplified CSP & specific CSP) and classifies them due to their origin (IT outsourcing (OC), virtualization (V), cloud computing (C)). If known, real world examples of security incidents resulting from a discussed problem are listed with short overview of counter measures.
In addition to operating issues, unclear laws and regulations are also a challenge for

No.	Issue	Origin	Incident examples	Affected security principles
A1	Misuse of administrator rights / malicious insiders	OC	Liebermann Password Survey (2011)	confidentiality, authenticity, authorisation, integrity, data protection, ac- countability, non- repudiation
A2	Missing transparency of applied security measures	OC & V	Heartland Data Breach (2009)	integrity, availability, data protection
A3	Missing transparency with security incidents	OC	Amazon Service Health Dashboard history (2011)	data protection, integrity, availability, non-repudiation
A4	Shared technology issues	OC, V	Red- & Blue- Pill Expolits (2008), Cloud Burst (2009), UDP- Flood Attack in (2009), Analysis of Amazon AMIs (2012)	integrity, availability, data protection, confidentiality, authentication, non-repudiation
A5	Data life cycle in case of provider switch or termination	OC	-	data protection, confidentiality
B1	Intransparent data location	C	-	data protection, confidentiality, availability
B2	Abuse and nefarious use of cloud resources	C	Zeus Botnet, Trojan and malicious office documents in Amazon EC2 (2010)	availability
B3	Missing Monitoring	C	-	non-repudiation, availability, data protection, confidentiality
B4	Insecure APIs	C	Attack of Amazon SOAP API, Gruschka (2009)	confidentiality, integrity, availability, non-repudiation, data protection, accountability
B5	Missing monitoring of cloud scalability	C	Miscalculated bill of costs in Amazon EC2, ToasterNET (2011)	A availability, accountability
B6	Missing interoperability of cloud provider	C	EMC Storage cloud closes (2010), Iron Mountain ends cloud storage service (2010)	availability
B7	Increased complexity	C	Amazon outage and data loss, (04/2011), (08/2011)	availability, integrity, confidentiality

Table 3.3: Overview of cloud security problems

industries' adoption of cloud technologies. IT Infrastructure Library (ITIL) [122] provides an auditable best practices catalogue for IT Service Management (ITSM). In addition, ISO Standard 27001:2005 [83] provides international auditable requirements for information security. ISO/IEC 27002 gives best practice recommendations on information security management. However, in cloud computing, IT resources are no longer solely in their own data center. Therefore, it is the discretion of a cloud provider

to follow ITSM and ISO standards. Germany's Federal Data Protection Act [119] specifies the acquisition, processing and storage of personal data. It is known to be the strictest data protection law within Europe. §4b (2) and (3) of the act define that personal data can only be transferred for processing into countries with the same adequate level of privacy protection laws. In addition, §4 (3) and §4 (16) of the act specify that whenever personal data are acquired and/or processed by third-party instances, the affected person has to be notified. Finally, §11 (1) states that where other bodies are commissioned to collect, process or use personal data, the responsibility for compliance with the provisions of this act and with other data protection provisions shall rest with the principal [119]. From the interpretation of this act, users must know the exact location of their data are and their cloud providers' court of jurisdiction. Unfortunately, this violates one of the cloud computing principles to host services wherever free resources are available.

Although existing recommendations, standards and laws provide well-established security and privacy rulesets for data center providers, it is becoming clear that they are not designed for virtual environments. Hence, a privacy and security framework is needed that proves the validity and applicability of existing laws to cloud computing.

3.5 Security Audits in Clouds

Since the literature review on existing cloud audit projects (Chapter 3.3) identified a strong demand in technically sound cloud specific audits, research was done to investigate, how security audits need to change to mitigate the identified cloud security issues. That's why, this section introduces different IT security audit types and discusses how classic audits need to change to consider the special characteristics of cloud computing environments and their security. Important challenges for cloud audits are

presented, and the main questions are given which a cloud audit should answer. This section finishes with a discussion of IT security audit industry standards for traditional data centres as well as new standards for cloud environments and a novel Cloud Audit Test Criteria catalogue gets introduced.

3.5.1 IT Security Audit Types

An audit can be defined as:

"*Formal inspection and verification to check whether a standard or set of guidelines is being followed, records are accurate, or efficiency and effectiveness targets are being met.*" [123]

The audit of IT environments focuses upon a particular technology area, for example, network infrastructure. Generally, IT audits can be characterised into four areas: general controls audits, application control audits, network/infrastructure audits and system development audits. The IT security audit focuses upon security issues of the whole IT infrastructure and can be defined as the process of IT risk analysis and vulnerability assessment. Figure 3.4 shows the process of a security audit: typical phases are definition, analysis, reporting, organisation and validation. It is good practice to use the results of the validation phase as additional input for a future audit. Typically, these audits are part of a quality management process to reduce the number of security holes.

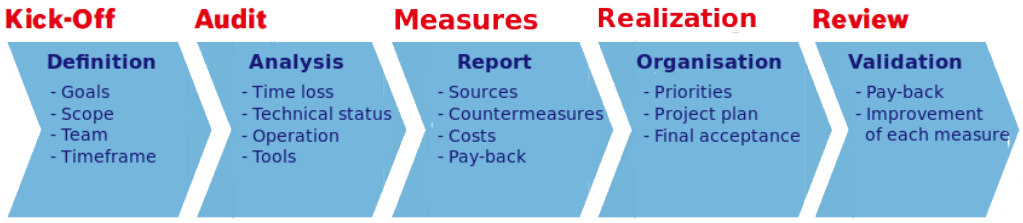


Figure 3.4: Traditional steps of an IT security audit process

IT security audits can be categorised into the following types:

Vulnerability assessment. Its task is to expose known security problems in all services of an IT infrastructure. Broad and automated vulnerability scans are used to assess the weaknesses of the IT. Experts manually verify detected vulnerabilities.

Vulnerability audit. It is a risk-based approach where IT is seen from the perspective of an attacker. It simulates an attack from malicious outsiders (hackers) by performing a penetration test. It is an intensive technical security audit with a high percentage of manual testing and verification.

Application security audit It is an intensive security audit of an application and its associated components (e.g. web application security scanner).

Vulnerability management. It specifies an automated vulnerability audit, and characteristics include automated, regular vulnerability scans and documentation of detected vulnerabilities in chronological order over multiple scans.

3.5.2 Classic IT Security Audits vs. Cloud Audits

For classical IT audits, today's standard is the Statements on Standards for Attestation Engagements No. 16 (SSAE 16) [124] report. SSAE 16 is an AICPA auditing standard for reporting on controls at service organisations (including data centres) in the United States. It requires that the auditor obtains a written assertion from management regarding the design and operating effectiveness of the controls being reviewed. This should minimise the following IT related risks, which are also applicable for cloud infrastructures:

- Loss of business focus of the service.
- Solutions failing to meet business and/or user requirements. The service is not performing as expected.
- Contractual discrepancies between the service user and the service provider.

- Compromised security and confidentiality.
- Invalid or incorrect processed transactions.
- Pure software quality (high number of failures).

With the appearance of cloud infrastructures, cloud-specific risks regarding IT security audits have been discussed and addressed by many researchers [76, 75], industry [87] and institutions [86, 74]. It can be summarised as:

Greater dependency on the provider Access to data or the control of resources in the cloud is still very much provider dependent. The cloud resource access interfaces are complex, and the extra control interfaces increase the vulnerability of cloud infrastructures. The risk of data lock-in is high, and because of the appearance of many new cloud providers, the risk of bankruptcy should not be neglected. There is a lack of standardized access interfaces to the cloud.

Increased complexity of compliance with laws and regulations Although a service is hosted at a cloud provider, the customer is still responsible for the data and service quality to the service users. Thus, the laws and regulations of a cloud provider country might be quite different than from the cloud customers'. The nature of cloud computing is to hide the location of the resources to the customer. The processing and data location can be anywhere, which might violate laws (e.g. European law of privacy forces the location in Europe for personal private data).

Reliance on the Internet The organisation's data stored in the cloud is only accessible through the Internet, which raises further security issues like data integrity, privacy and all kinds of attacks from this public environment.

Dynamic nature of cloud computing Processing and data location can be changed at any time because of load-balancing reasons or infrastructure failure. This causes many monitoring and controlling problems, and therefore, arguably the level of security decreases. Since the provider can scale the customer's infrastructure automati-

cally, the user must have control of this to limit the number of instances and control of the costs. Otherwise, a denial of service eats up all the revenue of the business service.

Thus, for cloud computing, an audit needs to clarify the following questions:

Privileged user access Since the provider has root access to the infrastructure and therefore can read unencrypted data on the cloud storage. So the number of administrators with root access should be minimised.

Regulatory compliance Customers are responsible for the data, even if it is in an external data centre. It has to be ensured that the provider takes care of backup, has reasonable data recovery times and strong encryption algorithms are used, if data encryption is needed.

Data confidentiality, integrity, privacy, availability and segregation In a cloud, the environment is typically shared among the customers. It is important to verify that it is secure. If the VM of another company is compromised, would my company VM be affected? Do you want to share a resource with your competitor? For many applications, resource sharing is acceptable, but for enterprise critical applications, you might want resources exclusively. Can a provider offer this? Special interest should be taken in understanding how the data is segregated and secured at the cloud provider. Is data replicated over multiple sites? Are backup strategies logically consistent? Is data really encrypted? Is data access limited to the customer's application? Is it possible to limit the data location to predefined areas? The cloud provider should transparently inform about the key management, access control, data segmentation, used encryption algorithms utilised, etc., of the cloud infrastructure. Additionally, business continuity plans and disaster recovery plans have to be defined in cooperation with the provider.

Investigative support Suppose the customer's resources are compromised. The

provider might have problems in undertaking forensic analysis, since logging in cloud environments are not user partitioned.

Monitor and control of cloud services Do customers get service level agreements, which can be adapted to the needs of the customers? Will the customers be able to monitor and manage them afterwards? Do the cloud interfaces offer sufficient and reliable information for the integration, control and monitoring tasks? How is data audited, which is stored, transmitted and processed outside the company? Is there access to accounting information?

Data retention For data stored in a cloud, questions need to be answered: How long can data be stored? How are data archived? How much is budgeted to retain data [125]? For retaining data from the cloud, it is important to clarify the following: How can data be retrieved? How is data integrity maintained during this process? How is data removed/securely wiped from the cloud storage systems?

Service level agreements are most often used to clarify the majority of these questions [1]. Nevertheless, SLAs are no support for a cloud customer without enforcement or traceability. It is important to provide a customer with the ability to check log data (physical, virtual and logical), event transport and storage services as well as event processing rules derived from SLAs.

From the technical point of view, the following challenges need to be covered [126]:

- **Loss of 1:1 mapping**: Due to the technology shifts towards VMs, virtual landscaped, location transparency is not clear for the customer.
- **Static gets variable**: Dynamic changes of IPs, data centres and servers depend on demand, time of day, etc.
- **Audit analysis**: How can data be retrieved, correlated and extracted meaningfully in a permanently changing infrastructure (VM start and stop)?

	Control environment/ company level controls	**Information security**	**IT service delivery/ operations**	**Systems development**	**Financial reporting system**	**Specific technologies or incremental requirements**
Best practices guidance	COBIT COSCO	ISO27002	ITIL ISO 20000-2	CMM/ISO 21827	ITGI-SOX	ISO var. ANSI var. NIST var.
Certification/ audit criteria/ requirement		ISO 27001	ISO 20000-1			
Regulatory/ Industry requirements		FFIEC HIPAA HITRUST NIST PCI ISO2700X			SOX PCAOB	EV SSL
Audit framework	SAS 70 SysTrust WebTrust BITS FISAP	PCAOB	WebTrust CA WebTrust EV GAPP			

Table 3.4: Industry standards for IT security [16]

- **Audit as a service**: For customers, it might be important to audit their business processes across multiple cloud providers.

Towards a Cloud Audit - Audit Standards

Multiple industry standards exist regarding compliance, regulation and best practices. Compliance to these standards enables companies to perform IT security audits which fit to their infrastructure. Since cloud infrastructures are definitely a special kind of IT infrastructure, cloud service providers (CSP) need to consider what IT services customers are allowed to run on their infrastructure and which industry standards apply to that business model. Table 3.4 shows available industry standards and their special focus [16]. Over the past three years, new IT security standards appeared, which are specialised for cloud infrastructures:

- CloudAudit A6: Automated Audit, Assertion, Assessment and Assurance API [127]
- EuroCloud Star Audit [95]
- Cloud Controls Matrix by Cloud Security Alliance [128]

CloudAudit A6 Its goal is to provide a common interface and namespace that allows cloud computing providers to automate the Audit, Assertion, Assessment and Assurance (A6) of their cloud environments. The interoperability between different clouds to avoid resource lock-in is important. It should be ensured that virtual machines can be controlled and hosted at different cloud sides. Therefore, the cloud provider should offer standardized interfaces to make the cloud more transparent in a secure and reliable way. One initiative is the DiffCloud interface, a language- independent REST-API.

EuroCloud Star Audit is a certificate for SaaS providers. It is the first specific certification for the Software as a Service model by the German EuroCloud Deutschland_eco e.V. [95]. The audit aims to establish a high level of security and transparency for users and providers alike. The audit starts with the provider's general profile; carries on with contract and compliance including data privacy protection, general security, operation and infrastructure and operation processes and goes as far as application and implementation. The audit consists mainly of six steps:

1. *Questionnaire:* The SaaS provider fills out a questionnaire about company profile, contract clauses, compliance, security and safety, infrastructure, business processes and implementation.

2. *Evaluation of questionnaire:* Auditors evaluate the questionnaire.

3. *Auditor interview:* Auditors interview the SaaS provider about questionnaire details, validity of certifications and implementation of documentation processes.

4. *On-site verification:* Auditors verify in an on-site visit questionnaire details, validity of certifications and documentation processes. This includes a visit of the provider's data centre if applicable.

5. *Evaluation and star ranking:* Auditors evaluate results based on a point-based evaluation matrix to decide which SaaS stars can be assigned. Detailed information about the matrix can be found in EuroCloud quick reference [129].

6. *Assignment of certificate:* The provider gets 1–5 SaaS EuroCloud stars assigned, dependent on the results of the evaluation. The certificate is valid for 24 months.

Although Eurostar Cloud Audits can already show some references of firms who successfully got the certificate [130] it remains unclear who the auditors are and how their qualification is verified.

Cloud Security Control Matrix Published by the Cloud Security Alliance, the cloud Security Control Matrix (CCM) is designed to provide fundamental security principles as guidance for cloud providers and to assist prospective cloud customers in assessing the overall security risk of a cloud provider. It provides an overview of audit attributes for a cloud infrastructure and classifies which cloud service models as well as cloud infrastructure components are affected by this attribute. It furthermore provides information about which specific section of available audit industry standards (as listed in Table 3.4) is addressing the respective issue.
The analysis on cloud security issues (Section 3.4) identified open research topics in the area of cloud computing security. The literature review on cloud audit projects (Chapter 3.3 as well as the analysis on audits (Section 3.5) showed, that they are a feasible approach to mitigate the identified problems. However, a cloud audit system needs to respect the cloud's characteristics, especially it flexibility and frequently changing infrastructure. There is clearly a need for a novel cloud audit system, since none exist so far. Thus, this research will continue on the development of a novel cloud audit system (to be presented in Chapter 4). To describe the main targets for the audit system to be developed, a well acknowledged approach in Software Engineering is the

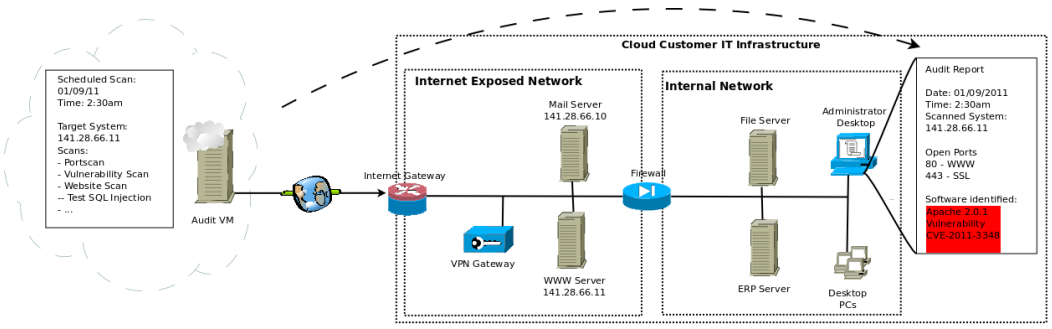


Figure 3.5: Audit from the Cloud

definition of use cases, to describe the (non-) functional, architectural and derived requirements and stakeholders of the target system.

3.5.3 Cloud Audit Use Cases

While cloud environments cause new challenges to traditional IT security audits due to their characteristics, they also enable new business cases to perform security audits on a regular basis. This section discusses the following possible use cases for cloud audits, which also form the use case scenarios for the work of this thesis:

a) Audit of non-cloud IT

b) Audit of cloud IT from the cloud customer point of view

c) Audit of cloud IT from the cloud provider point of view

Use Case A) - Audit of Non-Cloud IT

A typical enterprise is running at least the following basic IT infrastructure:

- File server to store documents
- Web server for company web site
- Mail server
- ERP system for financial transaction and reporting
- Internet connection and basic network services

Installation and maintenance are undertaken either by an external provider or an internal system administrator. Since some of these components are exposed to the internet due to their very nature, IT security audits should be performed to provide a descent level of protection of data and system availability. But especially small and medium enterprises face the following problems:

- Costs of an IT security audit performed by a third-party security provider are out of proportion to the company's revenue and available IT budget.
- Security is also undertaken by the company's administrator; however, frequently, IT security specific knowledge is missing. Priority is more commonly attributed to system maintenance and security controls rather than training.

Due to the cloud computing's pay on-demand model, "audits from the cloud" can be offered as depicted in Figure 3.5. A cloud customer can rent an "Audit VM", which was compiled by a security provider, including typical vulnerability assessment software. The tools are configured to start automatically in a logical order after the VM was booted, working through a list of target IP addresses of systems to be scanned. These will be the internet exposed systems of the customer. The results are conditioned in a standardized form to an audit report, which is sent to the customer's administrator. If security problems were identified (for example, an outdated version of web server software), recommendations, (for instance, from the Common Vulnerability and Exposures (CVE)) database are given on how the problem can be fixed. Customers subscribed to this service, schedule the scans to be performed either once or on a regular (for example, weekly) basis. Thus, common security problems like vulnerabilities due to outdated software, insecure configuration of services or compromised systems can be detected. Simultaneously, a comprehensible documentation of a system's state over time gets created. After the scan is completed, the report gets mailed to the customer and the audit VM gets shut down. Customer benefits are:

- Pay on-demand model: Audit VM only costs during runtime
- Security knowledge comes from external provider who maintains the audit VM
- Regular vulnerability assessment of Internet-exposed systems
- Audit report in standardized format provided taking into account a system's security status over time

It is imaginable that this service could be extended by scanning customer systems, which are not directly exposed to the Internet. Therefore, an authorised SSH host key or VPN certificate of the audit VM could be imported to an internal customer gateway, allowing the audit VM to first establish a connection to a customer's data centre. Then internal systems can be included in the scan as well, as depicted in Figure 3.5 There are already some companies on the market that offer a similar service to the described use case:

- Retina cloud by eEye Digital Security [131]
- The cloud Penetrator by SecPoint [132]
- Website security and anti virus scanner by Kyplex [133]

Use Case B) - Audit of Cloud IT from the Cloud Customer Point of View

In this use case, a cloud customer already uses a cloud offer and runs some instances (VMs) in a cloud. Due to the introduced cloud computing's characteristics and resulting problems, the customer faces the following problems:

- Missing monitoring of cloud instances
- Data security issues due to unknown data location and shared technology
- Missing auditability of the cloud provider due to missing transparency
- Loss of overview due to frequent infrastructure changes (VM start and stop)

In the traditional data centre scenario, the server landscape does not change often, and especially SME administrators know "their" systems by heart. In cloud computing, this can change due to the scalability of cloud resources. Dependent on the demand, the quantity of a customer's active cloud instances can increase and decrease quite frequently, for example, to fulfil a demand of service requests. Since cloud computing offers "inexhaustible" computing resources, users as well as administrators pick up on this advantage quite fast. For example, getting an additional machine exclusively just to try out a new version of a certain piece of software was very unlikely in traditional IT environments; in cloud computing, this is only a couple of mouse clicks away at little cost.[1] Administrators like this, because demand can be satisfied quite quickly, other running systems are not affected and there is no additional physical space needed. But this comfort can quickly lead to a loss of overview of the entire infrastructure, which is critical for securing it. Furthermore, in traditional data centres, security administrators harden systems and use a combination of firewall rules and intrusion detection system to secure it. But in the cloud, this is not applicable anymore due to the loss of control over hardware and shared technology issues. To overcome these problems in this use case, each cloud instance and the corresponding cloud infrastructure, for example, virtual switches, VM hosts, router and switches, are monitorable. Therefore, an agent framework can be used, providing "audit agents" deployed at core components of a cloud infrastructure, as illustrated in Figure 3.6. It shows the cloud reference architecture based on work from University of Los Angeles and IBM [48], presented earlier in Chapter 2. By adding audit agents to every layer, transparency to the cloud infrastructure can be provided for the user. Each agent is producing events in case an ominous transaction was detected. The cloud customer will define cloud security policies regulating, which components should be monitored

[1] If a private cloud scenario is considered, costs come down to zero in currency terms, and just the available resources count.

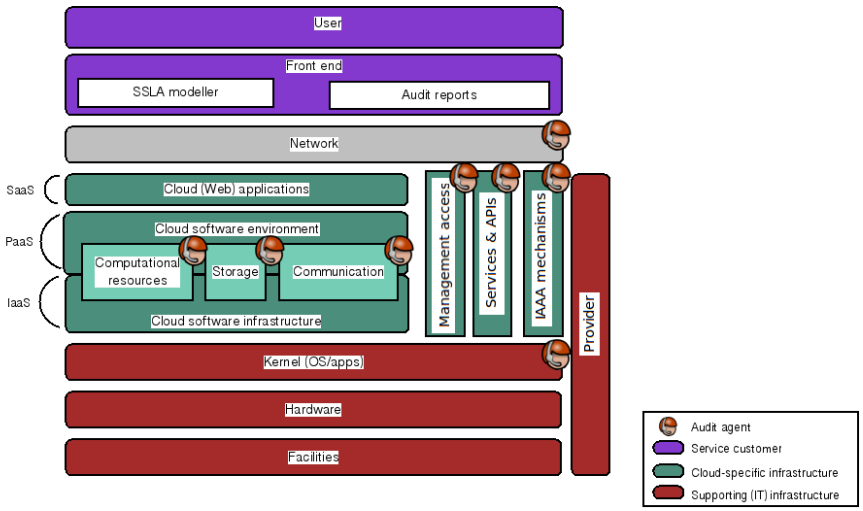


Figure 3.6: The cloud reference architecture extended by audit agents

and how, as well as alarm levels describing how the system automatically reacts in the event of a detected security incident.

Additionally, the described audit system from the previous use case can be applied to internal cloud instances as well, extended by using the audit agent's events as additional input for the audit report. The following advantages can be achieved for a cloud customer:

- Better overview of all customer-associated instances, possibly created from multiple accounts
- Transparency about cloud instances' security state
- Transparency about provider's administrative access

The following open-source or research projects are aiming to support this use case:

- CloudAudit A6 [127]
- Security Audit as a Service (SAaaS) [134]

Use Case C) - Audit of Cloud IT from the Cloud Provider Point of View

From the cloud provider's point of view, running and maintaining a cloud infrastructure are more challenging than a classic data centre. The reasons lie in cloud computing's characteristics, mainly its multi-tenant user model. To be successful, a cloud provider needs to prove the following:

- Compliance to laws, especially data protection laws
- Compliance to laws of all sub tractors
- Isolation and adequate segregation of shared computing and storage resources
- Measurements taken for availability, service and data protection, for example, backups and comprehensive continuity-of-operations plan
- Measurements taken to secure the cloud network environment, for example, intrusion detection systems, firewalls and logging facilities
- Accordance of cloud infrastructure with audit requirements
- Logging of all administrative access to customer's cloud resources, for example,two-factor authentication for cloud administrators, codes of conduct and confidentiality agreements
- Customer specific audit requirements

To fulfil this need, this research as well as governmental and industry security experts [88], for example, the German Federal Office for Information Security (BSI), recommends security audits and certificates as the preferred method of proof. Traditional IT security audits or penetration tests need to be adapted to a cloud's specific attributes, as described in the previous chapters. Principally, it is important to provide continuous monitoring of the cloud's security state over time. Due to the frequently changing infrastructure, the possibility that misusers of cloud resources are already

within the cloud's network (currently most are authenticated by a credit card number) are facts that traditional intrusion detection systems cannot cope with. Therefore, a monitoring system built on audit agents as described in use case *B) - Audit of Cloud IT from the Cloud Customer Point of View* can provide the following advantages for a cloud provider:

- Monitoring and detection of attacks against the cloud management system
- Monitoring of cloud usage behaviour to detect misuse of cloud resources (by legally registered cloud customer)
- Support for IT forensic investigations in case of successful attacks
- Displays security state of cloud infrastructure over time
- Proof of compliance to laws
- Possible interface to third-party security provider for external audits

Special Use Case: Auditing of Virtual Machine Images

A special use case for cloud audits is the security of virtual machine images. As listed in *Section 3.4 - Shared technology issues*, using third party appliance images from public marketplaces can pose a significant security risk. Therefore, not only running virtual machines need to be audited in a cloud environment, but also virtual appliance images, from which virtual machines are created. Thus, the topic gets elaborated in more detail in the following subsection. However, in the total scope of this research, auditing of virtual machine images is considered as either part of *Use Case B) - Audit of Cloud IT from the Cloud Customer Point of View* or *Use Case C) - Audit of Cloud IT from the Cloud Provider Point of View*.

VM Auditing Requirements

To be able to automatically audit VM images, it is essential to describe the security and privacy requirements, in a machine understandable way. This is commonly achieved by the definition of security policies, transferring a requirement into a checklist of one or multiple testable conditions. To respect cloud user's and provider's security requirements, both parties need to be able to create policies. A key factor for the success of such a system is the detailed and distinct definition of security policies. However, this is contrary to a short VM deployment process a cloud user expects. Therefore, this work proposes to create a very easily operable, security policy generator, where cloud users can define security policies in a human way of thinking. Simple policies could be supported by a graphical web interface with templates utilising check boxes or drop-down lists. This needs then to be translated into a machine understandable format, which results in the audit checks to be performed. The output of these checks needs to be translated back into a human understandable format, which will form the audit report submitted to the image creator, cloud provider and image user. In summary, the following important audit requirements can be identified:

- Automatic VM image auditing, to provide short response times to an image creator who wants to publish its image.
- The system needs to respect different security requirements from the image creator as well as the cloud provider.
- The system needs to produce a human understandable output in case an image did not pass the security check, providing the image creator with information about what prohibited the image release so that he is able to fix it.
- Security policies need to be described in a machine understandable way.

VM Audit Roles

When it comes to auditing virtual machine appliances, there are a couple of different roles, which need to be considered: *appliance user*, *appliance creator*, *cloud provider*, *audit service provider* and *audit tool provider*. These roles will be described in more detail in the following.

The **appliance user** is a customer of the cloud provider obtaining the virtual machine images via the appliance store. The main concern of the appliance user is to make sure, that the VM complies with the company's IT security policy when using third party appliances. Such a security policy may include the necessity of malware checks (e.g., viruses, trojans, spyware, rootkits etc.), checks for undesirable software (e.g., games, file sharing software) but also a more detailed view on the operating system and services configuration of an appliance. For example, if there exist unprivileged system user accounts for running a web server or if there are any leftover default passwords, which the appliance creator may have overlooked. The auditing's goal, from an appliance user's point of view, is to make sure a virtual appliance complies to his company's security policy, before the appliance is started and integrated in the company's IT infrastructure.

The **appliance creator** can be the cloud provider himself or a customer of the cloud provider. He creates individual VMs and shares them with other cloud customers using an appliance store. Before publishing virtual appliances, the appliance creator has to make sure that there is no private data, which could compromise privacy (e.g., logs, browser cache, user information like names and addresses), left on the image. Another, often overlooked, aspect are non-securely deleted files on the image's file system. It is often possible to recover such files with little effort using file carving tools, like extundelete [135] or winundelete [136]. The auditing's goal from the appliance creator's point of view is therefore mostly to make sure policies, which prevent the

disclosure of sensitive data, are used during auditing of the appliance before it gets published.

The **cloud provider** provides the technical infrastructure for running the virtual machine image and also runs the appliance store. Providers usually have little or no interest in restricting the creation and publication of virtual machine appliances, as long as there is no violation of laws or terms of use. Such violations may include the intentional distribution of malware, intentionally misconfigured services or any form of illegal content, such as pirated software.

The two remaining roles **audit service provider** and **audit tool provider** have no immediate interest in auditing virtual machine appliances. They merely complete the auditing process by providing additional services and tools. The audit tool provider designs, develops and provides programs and services for auditing virtual machine images. The audit service provider is a specialist in auditing IT infrastructure and therefore has extensive knowledge about auditing procedures and methodologies, which he offers to the cloud provider. They also provide the cloud provider with work-flows, recommendations about the tools to use, knowledge about currently emerging threats to security and privacy as well as any additional auditing know-how.

Auditing Categories

The auditing categories identified by the author are **security**, **privacy** and **legal concerns**. Multiple audit cases from these categories can be arbitrarily combined to form an auditing policy. Each of the previously described roles has a different view on the requirements the audit process has to fulfil. Table 3.5 illustrates this circumstance. The security category includes requirements regarding the absence of malware and otherwise undesirable software in the virtual appliance. Also, the preconfiguration of the appliance's services, like access rights on a file system level, the combination of services (e.g., mail daemons and network attached storage (NAS) service on the same

		Appliance User	Appliance Creator	Cloud Provider
Security	Malware	x		x
	Undesirable Software	x		
	Account Requirements	x		
	Login Requirements	x		
	Password Strength	x		
	Access Rights	x		x
	Service Misconfiguration	x		x
	Unwanted Service Combination	x		
Privacy	Browser Caches		x	
	Log Files		x	
	History Files		x	
	Insecurely Deleted Files		x	
Legal	Software Licenses	x	x	
	Illegal Content	x	x	x
	Customer Specific Requirements	x		

Table 3.5: Virtual Appliance Audit Categories

appliance), insecure default service configurations, the use of insecure default passwords and login requirements (allowing remote administrator access with passwords) are the most common concerns. The privacy category includes mostly requirements, which should help preventing unintentional data loss. This includes leftovers from the appliance setup process, like log files, command line history or insecurely deleted files. Additionally, this category also includes data generated by end-user applications (e.g., browser caches). The legal category includes all sorts of compliance requirements, for example illegal content stored on the image.

The appliance user obviously has much more interest in secure and compliant appliances. One could argue that data loss while using the appliance because of misconfigured services or backdoor programs could be assigned to the privacy category, but

this data loss arises because of security problems.
The appliance creator has mostly privacy concerns, when publishing virtual appliances. Checks for the previously described problems need to be thoroughly executed before publishing a virtual appliance. Also, not publishing preinstalled software licenses is a legal problem, which needs to be checked.
The cloud provider's main concern is protecting his own infrastructure. Therefore, the categories, which apply are security and legal. Checking virtual appliance images for malware may reduce the risk of malware spreading and hackers using the appliance store as a basis for their attacks. Checking for illegal content may also be necessary because the cloud provider stores virtual appliances.

As mentioned, in the total scope of this research work, auditing of virtual machine images is considered a sup part of Use Case B & C) - Audit of cloud IT from either the cloud customer or cloud provider point of view.

3.5.4 Cloud Audit Test Criteria Catalogue

As elaborated, security audits are a commonly acknowledged measure to validate security for IT infrastructures. However, since cloud computing introduces new security risks, cloud security audits need to consider the special characteristics of cloud computing. Based on the studies on cloud security issues (Section 3.4) and cloud audits (Section 3.5) multiple interviews and meetings with a German cloud provider [137] and a German IT security provider [138] were held to identify security concerns and challenges of applied cloud computing technology. As a result, an "Audit Test Criteria for Cloud Infrastructures" catalogue [9] was developed by this research. It is written in German (since it is also an official deliverable by the SAaaS project, which is funded by the German government), publicly available for download at the Security Audit as a Service project web site [139]. The catalogue is also an official deliverable of the

funded SAaaS research project, introduced in Chapter 1.4.

The catalogue is targeted for professional, safety aware cloud customer, IT decision makers who consider moving services to or changing their data centre structure to become a cloud provider and cloud computing provider and their administrators who wants to operate a secure cloud environment. Furthermore, it is also meant for security service providers, who investigate cloud infrastructures in a penetration test or a infrastructure audit. The catalogue provides over 140 test criteria. Each test criterion is classified into an audit category, numbered and sorted in ascending order. For example, the first audit test criterion (in German: **A**udit**p**rüf**k**riterium PK) "A1PK1 - Log analysis" is the first test criterion of the audit criteria "A1 - Security of network connections".

All criteria are classified (A, AA, or AAA) according to the official "protection requirements classes", defined in the IT security baseline catalogues [101] from the Federal Office for Information Security, Germany. Audit test criteria of class A apply for systems, data or usage scenarios with normal protection requirements. Class AA applies for systems, data or usage scenarios with high protection requirements, and class AAA apply for systems, data or usage scenarios with very high protection requirements. Each of the 140 audit test criteria is explained in detail within the catalogue. At the end of each subcategory a table (see Figure 3.7) summarizes all test criteria of the category to provide a quick overview. The protection requirement class is indicated in column 3 - 5.

All test criteria are divided in the following subcategories:

User Access: contains audit criteria to validate a secure user access to the rented cloud services, including programming interfaces, which are quite commonly used in IaaS and PaaS Clouds.

Network & Peripherals: covers requirements for typical network components of a cloud infrastructure, such as switches and routers, especially virtual switches, virtual

PK-No.	Description	Category		
		A	AA	AAA
A27PK1	Contract version	✓		
A28PK1	Existance of a formal defined security management process	✓		
A28PK2	Reporting about undertaken security measures	✓		
A28PK3	Regular execution of security audits	✓		
A29PK1	Compliance to guaranteed service recovery times		✓	
A30PK1	Existence of an official data protection supervisor	✓		
A30PK2	Existence of a formal data protection plan	✓		
A31PK1	Information about involved countries	✓		
A31PK2	Comprehensibility of all data processing systems		✓	
A32PK1	Logging and disclosure about governmental access		✓	
A33PK1	Logging and disclosure about security incidents at shared resources		✓	
A33PK2	Supply of relevant log data in case of security incidents		✓	
A34PK1	Information about key creation environment	✓		
A34PK2	Purpose of access keys	✓		
A34PK3	Storage of access keys			✓
A34PK4	Redundant storage of access keys	✓		
A34PK5	Secure deployment of access keys	✓		
A34PK6	Access to keys by CSP administrative personell	✓		
A34PK7	Protection of key management		✓	
A34PK8	Key archiving		✓	
A34PK9	Secure wiping of keys		✓	
A35PK1	Data export validation	✓		
A36PK1	Existence of a customer overspanning incident detection system		✓	

Figure 3.7: Summary audit test cases “Infrastructure overlapping criteria” [9]

networks and their segregation. Tests include the validation that only encrypted connections are used between cloud components.

Cloud Management System: This category concentrates on the security of the core component of an IaaS Cloud, the cloud management system. It is used for management and life cycle of the virtual resources offered to cloud users. Furthermore, it provides monitoring information about the cloud infrastructure and their services, such as load balancer, scalability thresholds or scalability actions. It is also important for user management, billing and accounting due to a cloud’s pay on demand characteristic.

Cloud Hosts: contains audit criteria on the security of cloud service hosting machines. Beneath a secure configuration of the hardware and its operating system it concentrates on the hypervisor security, which provides virtualization functions.

Cloud Backend Storage System: stores VM images (IaaS, SaaS) and user data of Storage as a Service services. Typically distributed databases are used for this in a cloud infrastructure. An important test critera is a proof of data security and secure data wiping in case of shared storage units (especially duplicates) or if a customer quits its cloud usage.

Infrastructure Overlapping Criteria: cover non-technical, organisational and contractional audit test criteria. Very important are service recovery guarantees regulating, at what time cloud resources are available again after an infrastructure failure. Test criteria in this category also oblige the provider to provide information about processes in case of governmental requests on customer data. Who is legally allowed to file a request on customer data disclosure? Is governmental access on customer date recorded and accessible to the cloud customer? This category covers also the proof of physical data location. In case of bankruptcy of the cloud service provider clear procedures must exist, on how customer get their data out of the cloud. Do export interfaces exist? Is the export format standardized or compatible to other cloud service provider? Finally, which technical measurements are taken by the CSP to detect cloud specific security incidents? Are early detection measures in place to protect parts of a cloud in case of a security incident?

Administrative Access from cloud service provider's personell: Since a cloud provider's personell, especially administrators of cloud infrastructure components, has privileged access to confidential data, transparency about access monitoring and traceability is very important.

IaaS specific criteria: covers all technical and organisational criteria which do not fit in any of the other categories, since they are specific to IaaS cloud infrastructures. In IaaS it is important to differentiate if VM images are provided by the CSP or completely created and managed by the cloud user. If VM images are provided by

the CSP, responsibility about compliance to the audit test criteria of this category is with the provider, whereas if no images are provided, responsibility lies with the cloud user.

PaaS specific criteria: Fundamental system settings of the underlying cloud infrastructure usually can't be managed by the cloud user. Thus, administrative responsibility is with the cloud service provider. Since in PaaS cloud applications are developed by cloud user, responsibility about security in data processing and management within the application is with the cloud user [140]. This subcategory contains PaaS specific audit test criteria.

SaaS specific critera: For SaaS, all IaaS and PaaS criteria apply. In addition, this subcategory lists SaaS specific audit test cases, such as specific export formats of user data in case of a provider switch or what measurements exist to detect misusage of the rented software.

Cloud Audit Criteria Excel Spreadsheet

Additionally to the catalogue, an Excel spreadsheet, which contains all audit test criteria was created as a result of this research. Figure 3.8 shows a cutout of the spreadsheet.

Requirement	Test criteria number	Criterion	Description	Protection Category			Category	Audit Area
				A	AA	AAA		
Filter	Filter			Filter	Filter	Filter	Filter	Filter
	A18PK6	Frequent validation for exiting security updates	Is software checked regularly for security updates (at least once per day)?	x			Cloud-Hosts	Monitoring
A19	A19PK1	Status hypervisor updates	Is the utilised hypervisor software up to date	x			Cloud-Hosts	Technology & Operation
	A19PK2	Usage of certified hypervisor software	Is a certified hypervisor software used (Minimum requirement: Common Criteria EAL 4)?			x	Cloud-Hosts	Technology & Operation
A20	A20PK1	Logging of all operations	Are all operations logged (Image-creation -duplication, deletion)?	x			Cloud Backend-Storage-System	Monitoring
	A20PK2	Availability of access logs	Are access logs available to customers?		x		Cloud Backend-Storage-System	Monitoring
A21	A21PK1	Security Updates	Are all available security updates for the storage software installed?	x			Cloud Backend-Storage-System	Technology & Operation
A22	A22PK1	Seperate storage of provider- and customer images	Are provider- and customer images stored seperately?	x			Cloud Backend-Storage-System	Technology & Operation
	A22PK2	User access	Is access on other customer data possible?	x			Cloud Backend-Storage-System	Technology & Operation
A23	A23PK1	Cloud storage overview	It is visible to a customer which data are stored in the cloud?		x		Cloud Backend-Storage-System	Monitoring
	A23PK2	Creation of a process directory	Does a customer gets all information necessary for creation of a process directory?		x		Cloud Backend-Storage-System	Organisation & Contract

Figure 3.8: Cutout of audit test criteria catalogue spreadsheet [10]

It can be used to provide input criteria based on the audit criteria catalogue for an

expert system. It is separated into the following columns: audit criteria, description, protection class, criteria class and audit area.

The *audit criteria* describes the technical or organisational measure which needs to be audited, such as: "Which measurements exist to test virtual machines on the availability of software updates (OS and applications) on a regular basis" (Figure 3.8, row 1). The *description* column lists detailed test information on the corresponding test criteria, such as possible tools for the test or minimum or maximum values. In this example it states: "Is a check for security updates performed at least once a day for all software installed on VM images?". The *protection class* column indicates if this test criterion applies for assets with a normal (class A), high (class AA) or very high (class AAA) protection requirement class. The *audit area* describes typical topic areas of security audits, such as technical components and operation, monitoring, organisation and contract or business continuity management. The spreadsheet enables user to filter the criteria corresponding to their needs, such as protection class of their assets, audit class or audit area. The catalogue is already used in industry, for example by the management of the data centre of the federal state Rhineland-Palatinate [141]. Further cooperative developments of the catalogue are planed.

3.6 Summary

This chapter introduced cloud security incidents to show that there is a demand for research on cloud security. Then, the main differences between traditional IT outsourcing and cloud computing were discussed. Amplified Cloud Security Problems (amplified CSP) and Specific Cloud Security Problems (specific CSP) were presented and a classification of affected core principles of information security was given for every identified problem. Table 3.3 summarised the presented cloud security problems, classified them according to their origin (IT outsourcing, virtualization or cloud

computing). If known, real-world examples of security incidents resulting from a discussed problem were listed, and a short overview of countermeasures was provided. Due to the identified lack of technical audit criteria for IaaS clouds (Chapter 3.3 an analysis of traditional it security audits was performed. To provide audits for cloud computing infrastructures, there is a critical need for audit criteria respecting cloud's characteristics and nature. The following limitations current audits and related work on cloud audits were identified:

- Lack of technical and organisational detailed audit criteria for clouds, especially IaaS clouds
- High fluctuation in infrastructure: In IaaS cloud infrastructure changes very frequently. An audit needs to respect this, re-validating security status after change
- Loss of 1:1 mapping: Due to the technology shifts towards VMs, virtual landscaped, location transparency is not clear for the customer.
- Static gets variable: Dynamic changes of IPs, data centres and servers depend on demand, time of day, etc.
- Audit analysis: Data needs to be retrieved, correlated and extracted meaningfully in a permanently changing infrastructure
- Audit as a service: For customers, it might be important to audit their business processes across multiple cloud providers.

To mitigate the first limitation, an audit test criteria catalogue for cloud infrastructures was developed and published. It contains over 140 audit test criteria, arranged in ten categories (depicted in Figure 3.9) to mitigate the elaborated cloud security risks. To address the remaining limitations, this research then went on to develop a Security Audit as a Service system for cloud computing to provide the following advantages

over traditional audits:

- High fluctuation in infrastructure: Consideration of a cloud's flexible nature
- Consideration of virtualization technology and their characteristics
- Consideration of a cloud's variability: Dynamic changes of IPs, data centres and servers of virtual instances
- Audit analysis: Meaningful correlation and presentation of data for cloud customers and providers
- Audit as a service: on-demand provider independent audit system, configurable in a flexible manner

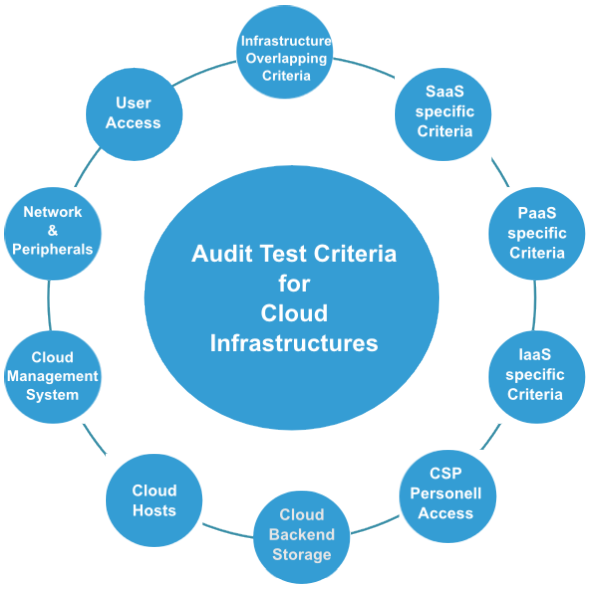


Figure 3.9: Audit Test Criteria Catalogue for Cloud Infrastructures [9] categories

4 Security Audit as a Service (SAaaS)

"Logic will get you from A to B. Imagination will take you everywhere."

(Albert Einstein, Theoretical physicist)

The previous work identified a need for a security system respecting cloud specific characteristics. Whereas organisational and regulatory needs could be satisfied with the previously presented audit catalogue, there is still a need for a technical audit solution for clouds. Thus, this chapter introduces the Security Audit as a Service (SAaaS) architecture. It is built upon intelligent software agents, which are aware of underlying business driven intercommunication of cloud services. This enables the architecture to be flexible and to supported cross customer event monitoring within a cloud infrastructure. An agent definition is given, and it is elaborated how agents can improve incident detection and auditing in a highly dynamic infrastructure. Finally, the SAaaS architecture components is presented in detail. It forms the second novel contribution of the research.

4.1 Introduction

Monitoring and auditing the security of large IT infrastructures with distributed sensors as input feeds is commonly done by Intrusion Detection Systems. But this approach breaks down for cloud infrastructures, mainly because of the complexity and frequently changing environment driven by the user. Literature review on cloud audit systems (Section 4.2) shows, that there has been little effort made to address this issue of auditing a frequently changing cloud infrastructure. To mitigate this problem, this research proposes a concurrent audit system for cloud infrastructures: Security Audit as a Service. It is a novel audit system which can be either used as an on-demand audit system from the cloud, which can evaluate an user's internet exposed IT components, or as a concurrent audit system, which monitors cloud instances, reacts on changes within a cloud infrastructure and automatically performs security audits of the affected cloud instances to evaluate their security status. This is called "concurrent auditing". To achieve this, SAaaS is build upon intelligent agents, collecting data directly at the source, analysing and aggregating information and distributing them, while considering the underlying business process. The usage of agents enables a concurrent auditing of cloud components, while maintaining the cloud's specific flexibility.

Summary of Research

This part of the research develops the Security Audit as a Service system, which fulfils the identified limitations on cloud audits, by providing:

- A virtualization and cloud software independent audit system
- A cloud audit system, which respects IaaS cloud characteristics
- A possibility for specifically targeted audits of affected cloud instances
- A lightweight agent framework to respect a cloud flexibility and scalability

Parts of this research phase have been published in the following papers and presented to the research community at the corresponding conferences:

- An Autonomous Agent Based Incident Detection System for Cloud Environments, *3rd IEEE International Conference on cloud Computing Technology and Science 2011* [142]
- Incident Detection for Cloud Environments, *International Conference on Emerging Network Intelligence 2011* [143] - **Winner of Best Paper Award - Track Security**
- Validating Cloud Infrastructure Changes by Cloud Audits, *Eighth IEEE World Congress on Services 2012* [144] - **Winner of 2nd place at IEEE Services Cup 2012**
- An agent based business aware incident detection system for cloud environments, *Journal of cloud Computing: Advances, Systems and Applications 2012* [145]

4.2 Related Work

Before presenting the Security Audit as a Service architecture, related work on IaaS cloud security systems utilising agents is elaborated. Raj et al. [146] introduce a virtualization service implemented as Xen Virtual Machine extensions, which provides role based access control based on a trust value of a VM. This trust is based upon a VM's attributes, such as number of open network connections. Access to different cloud services, for example file access is given on a VM's trust value. The implementation presented is mainly based on Xen tools. The proposed services by [146] are valuable, however strictly limited to Xen. Thus, a technology independent audit system is needed. The developed SAaaS system of this research will deliver a hypervisor technology independent audit solution, which is widely applicable.

In [147] Wei et al. present a VM image management system, which controls VM image access and tracks image provenance to address the issue of security patches for VM base images. It provides a prototype image scanner to scan software versions installed within a VM image and filter for the user to exclude images with unpatched software stacks. The proposed SAaaS architecture will extend the presented trust model of [147] by the ability to evaluate business process driven cloud infrastructure changes by the evaluation of underlying security business flows.

To provide secure cloud hosts running VM images the authors of [148] and [75] propose the usage of TPM/TCPA crypt chips for secure OS installation. The SAaaS architecture can utilise all of the above introduced techniques to establish a trusted computing base for cloud environments and extends them to provide cross customer trust.

The applicability of using complex event processing (CEP) in cloud environments is demonstrated by Schaaf et al. in [149]. The authors present a predictive cloud broker which reacts to changes in business processes and reflects them in up- or downscaling of cloud resources (VMs). As a decision base distributed monitoring instances in relational database systems are used to feed the CEP engine. While this approach is an initial step in the same direction as this research, the work remains quite high level and in particular focuses upon business process changes whereas SAaaS targets the detection of security incidents. Despite these different application areas, principles of this work can be applied to SAaaS. It extends them by the consideration of business process flows, and a reduction of event storms.

Due to the loss of hardware governance in cloud computing, customers need to trust the provider that data does only get stored on the providers' storage compliant to applying data protection laws. This can result in unclear data location. Ries et al. present in [150] a geo-location approach based on network coordinate systems and evaluate the accuracy of three prevalent systems.

Detecting abuse and nefarious use of cloud resources can be a challenging task, which proves the rather high level work of [149]. Just evaluating VM usage data, like CPU and memory usage, or the number of open network connections is not enough, since it can result in a false-positive decision. To filter out nefarious use of a cloud infrastructure, an audit system has to combine usage and network data of the cloud wide network. Due to the distributed nature of cloud computing, information about network flow has to be collected at many different physical locations. However, to get the whole picture this data has to be analysed in the overall context. Therefore, for the audit system of this research, distributed sensor agents get developed, which feed an anomaly detection system to evaluate the cloud's security status in an overall context. Furthermore, an anomaly detection system gets developed, which utilises behaviour analysis and anomaly detection techniques to distinguish between "normal" and nefarious use of cloud resources.

Distributed Agents Research

Since a core component of the Security Audit as a Service architecture are distributed agents (presented in Chapter 4), this section presents related work on the usage of agent architectures.

The advantages of using agents to overcome the challenges of monitoring a frequently changing infrastructure is discussed in research especially in the area of Intrusion Detection & Prevention Systems (IDS, IPS) and demonstrated in [151, 152, 153, 154]. Lui et al. demonstrate in [151] that anomaly detection can enhance the detection of new, unknown network attacks in an IDS. They show that certain data mining algorithms are more suited to detected certain network attacks than others. Therefore, multiple agents are proposed, implementing different data mining algorithms to lead to a higher, more precise detection rate of unknown network attacks. This supports

the approach of this research, that agents can enhance anomaly detection of unknown attacks. While the agents presented by Lui et al. are very static in their perimeter, the proposed SAaaS agents in our work can be extended during runtime to adapt better to the current infrastructure state.

Zamboni et al. present in [155], how traditional Intrusion Detection Systems can be enhanced by using autonomous agents. They confirm the advantages of using autonomous agents in regards to scalability and system overlapping security event detection. In contrast to the SAaaS architecture, their research is focusing on the detection of intrusions into a relatively closed environment, whereas this work applies to an open (cloud) environment, where incidents like abuse of resources needs to be detected. While Chirumamilla et al. show in [152], that agents can enhance the security of wireless networks they don't really benefit from typical agent characteristics like deployment on demand as the SAaaS agents presented by this research.

Mo et al. introduce in [156] an IDS based on distributed agents using mobile technology. They show, how mobile agents can support anomaly detection thereby overcoming the flaws of traditional intrusion detection in accuracy and performance. The paradigm of cooperating distributed autonomous agents and its corresponding advantages for IDS' is also shown by Sengupta et al. in [157]. The presented advantages apply for the SAaaS agents, developed in this research as well.

Krügel and Toth show in [154], that processing limitations of IDS can be enhanced by using mobile agents. Instead of moving sensor information to a central processing unit they use agents to correlate the monitored event data, thus increasing fault tolerance and scalability of the intrusion detection system. This idea is partially supported by the event preprocessing of the presented SAaaS agents in this work. The concepts of agent frameworks have been widely used in distributed networking environments. Due to their characteristics (elaborated in Section 4.6) they are perfectly suitable to fulfil

requirements of a cloud security audit system. However, due to the best knowledge of the author no research so far has utilised them to mitigate cloud security issues as it is done by this research.

4.3 SAaaS Use Cases

The Security Audit as a Service system utilises the following three use cases to show the usefulness of the proposed architecture. They are a more specific version of the cloud audit use cases described in *Chapter 3.5.3 - Cloud Audit Use Cases*.

Use Case 1 - Automated security audit: In this use case, the SAaaS architecture is used as a Software as a Service solution. An audit can consist of regular vulnerability scans of a user's internet exposed systems (not necessarily cloud instances). Results get automatically evaluated, post-processed and submitted as a security report in a standardised format to the user. Additionally, to simplify black box scans, it is imaginable to deposit an entry credential (e.g., a ssh key pair) in the service so that the service can log in and perform internal security scans. While such systems already exist as appliances (e.g., Nessus appliance), especially small SMEs can profit from this service running in a cloud since they only need to pay per scan. For the cloud provider this service is valuable since computing resources are only allocated for the duration of a scan. Afterwards, the compute resources are released and made available for different tasks.

Use Case 2 - Auditing of cloud VM images Using third party appliance images from public marketplaces can pose a significant security risk, as introduced in *Chapter 3.4.2 - A4 Shared Technology Issues*. Therefore, images from which virtual machines are created, need to be audited. This is important for cloud provider to protect the cloud infrastructure turning into a "Swiss cheese" by running many vulnerable virtual machines. It is also important for the cloud user to protect their virtual instances,

since data could be stolen or the VMs get used as an entry point to their own data centre.

Use Case 3 - Monitoring and audit of cloud instances: User VMs running in a cloud infrastructure are equipped with a SAaaS agent. The user creates security policies defining the behaviour of this VM to be considered "normal", which VM components are to be monitored and how to alert the customer in case of system deviation from the defined manner. The status gets conditioned in a user-friendly format in a web portal - the SAaaS security dashboard. This continuous monitoring creates transparency about the security status of a user's cloud VMs, hence increasing the user's trust into the cloud environment.

Use Case 4 - Cloud infrastructure monitoring and audit: The security status of the entire cloud environment, especially the cloud management system, access to customer data and data paths are monitored. Usage and communication behaviour profiles are created automatically and continuously analyzed for substantial changes. This way, monitoring across different customers is used by the cloud provider as well as a third parties, like a security service provider to ensure the overall cloud security status. Standardised interfaces enable security audits of the cloud infrastructure, which can lead to a cloud security certification.

Before explaining the SAaaS architecture in detail, this chapter continues explaining the underlying SAaaS event processing sequence. To support this, consider the following example of a typical web application architecture, consisting of one or multiple web servers and a database backend is deployed over multiple VMs in a Cloud. The VMs are logically grouped together, and labelled as *WWW-Cluster1*. Initially, the cloud customer's administrator installs the VMs with the necessary software, e.g., Apache web server, MySQL database. After the functional configuration, security policies are modelled to describe the target infrastructure state. This can be:

A) Technical rules, such as allowed network protocols and connections between VMs.

A finished configuration of the web servers and a notification trigger in case changes to its config files are recognised. As a result of these policies, software agents called `VM agents` are configured with the necessary tools to monitor and assess these requirements and automatically deployed to the VMs.
B) Business flow related security policies can be created as well, such as a simple scalability policy: "If the cloud management systems gets an upscale event request for components of *WWW-Cluster1*, first the actual load of all web servers needs to be checked. If the average load over all web servers is not higher than a certain threshold, e.g., 100 http connections / web server, upscaling gets denied and an alarm is raised since the event must originate from a systems failure or a security incident at the cloud management system. The same scenario works for downscaling in an inverse manner: If a downscaling event for *WWW-Cluster1* gets detected, but the actual load is above a downscale threshold, an alarm gets raised.

4.4 Cloud Audits Using Agents

To generalise this scenario: in the SAaaS architecture a modelled security state of certain components gets monitored by agents, which are deployed at the specific resource, such as the cloud management system, a cloud host or a VM. Agents were chosen over plain objects because of their following attributes, according to [158]:

- Agents function continuously and autonomously in a particular environment
- Agents are able to carry out activities in a flexible and intelligent manner that is responsive to changes in the environment
- Agents are able to learn from its surroundings
- Agents can communicate and cooperate with each other
- Agents can be (re-) used by multiple entities

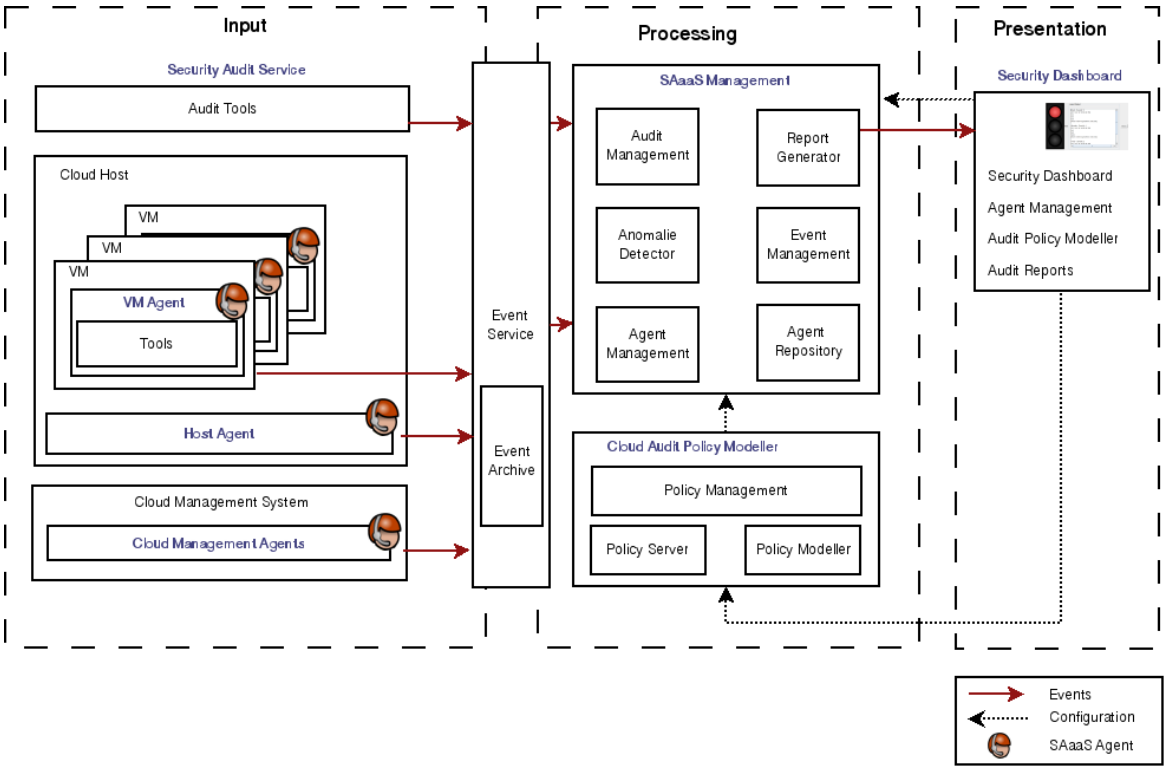


Figure 4.1: SAaaS event processing sequence

Since the agents in the SAaaS architecture are running independently, not necessarily connected to a certain central instance, agents can receive data from other instances and send information to other instances like other SAaaS agents or the SAaaS' event processing system. The "central" event processing system gets itself implemented as an agent, which can be scaled and distributed over multiple VMs.

4.5 SAaaS Event Processing

Figure 4.1 gives a high level overview of the SAaaS architecture and where events are generated and forwarded within the SAaaS architecture. It can be divided into three logical layers: Input, Processing and Presentation.

Input: The SAaaS architecture gets its monitoring information from distributed agents, which are positioned at key points of a cloud's infrastructure, to detect changes

in a cloud environment. Possible key points are: running VMs of cloud users, the VM hosting systems (Cloud hosts), data storage, network transition points like virtual switches, hardware switches, firewalls, and especially the cloud management system. An agent integrates several monitor and policy enforcing tools. A logging component is recording the chronological sequence of events produced by the agent, building audit trails.

Processing: Each SAaaS agent receives security and audit policies from the audit policy modeller component. Through security policies, each agent gets a rule set (its intelligence) specifying actions in case of a cloud change (e.g., modification of a frozen config file). Thus, every occurrence gets first preprocessed by an agent, which reduces communication between VM agents and cloud management agent. The Cloud Audit Policy Modeller consists of a graphical Policy Modeller (policy front end) and a Policy Server (policy backend). The policy modeller is used to create cloud specific audit policies. An example could be: "In case of a registered VM upscaling event, the minimum load on all web server VMs must be at least > 1000 *requests/second*". Details on cloud audit policies are discussed in Chapter 5. Cloud audit policies get sent from the Policy Management to the Agent Management to configure the corresponding agents. Different agent templates are stored in an agent repository and created on demand. By using the monitoring information of the distributed agents in combination with the security policies, a cloud behaviour model is built up for every cloud user. This is analysed by the Anomaly Detection module to detect user specific or user overlapping security incidents. Countermeasures can then be applied to early detect and prohibit security or privacy breaches. Details on the Anomaly Detection are presented in *Chapter 6 - Anomaly Detection*. An Audit Management module manages concurrent audits of cloud instances, dependent on the events received from the SAaaS agents or the anomaly detection module. The Report Generator conditions events, corresponding security status as well as audit report results in a human friendly presentation.

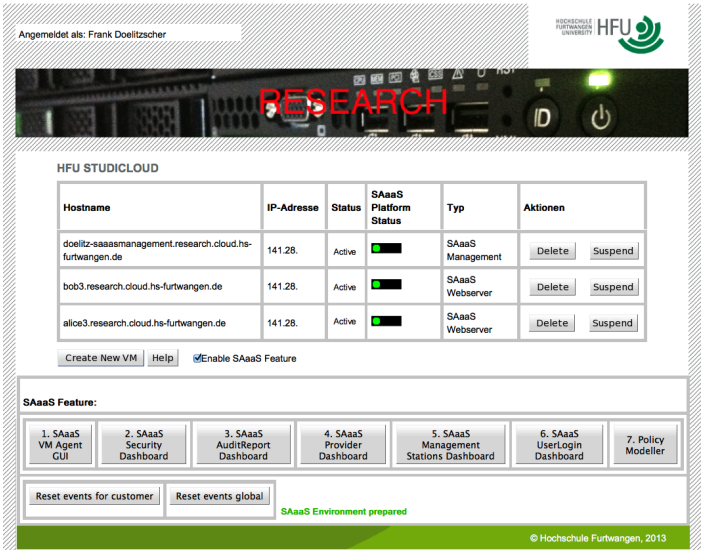


Figure 4.2: Security Audit as a Service Management Interface

Presentation: As a single interaction point to cloud users, the SAaaS Security Dashboard provides VM information, audit reports and cloud instances' security status. Information is organised in different granular hierarchies, depending on the information detail necessary. At the highest level a simple three colour indicator informs about a user's cloud services overall status. It also provides a graphical user interface to deploy agents to cloud instances or allows user to manually trigger a specific audit. Audit policies can be created through the Policy Modeller. Figure 4.2 shows a part of the security dashboard prototype, which gets described in more detail in Chapter 7. Communication between the distributed agents and the security dashboard is handled by an Event Service. Events are stored in an Event Archive.

4.6 SAaaS Agents

Agents provide the basis of the SAaaS architecture's main information framework. As a correlating technology, distributed objects could also be a suitable technology for the targeted audit system. A lot of discussion has been done, how agents differ

from normal, distributed programs [11]. However, in comparison agents provide the following advantages over distributed objects:

Standardised design Whereas a plain distributed object approach establishes the need for developing a custom deploy and communication framework, utilising an agent framework provides already well defined program structures, interfaces, a communication language and separation of duties. This reduces the risk of unnecessary complexity and beginner's mistakes.

Asynchronous and autonomous execution Distributed objects normally rely on a continuously open network connection between the central server and the dispatched object. In case of a security environment, such as the targeted sensor agents in the SAaaS framework, it is important that sensors are continue in their operation even if the network connection is disrupted, since this could also be intentionally done to hide a security incident. Tasks can be embodied into agents and transferred to their target execution environment, e.g. a VM. After being deployed, the agent becomes independent of its creating process, see Figure 4.3. It can act asynchronously and autonomously continue and sharing its information with other entities (agents), even if the connection to a central instance is disrupted. It can reconnect to a central instance when network connectivity is re-established.

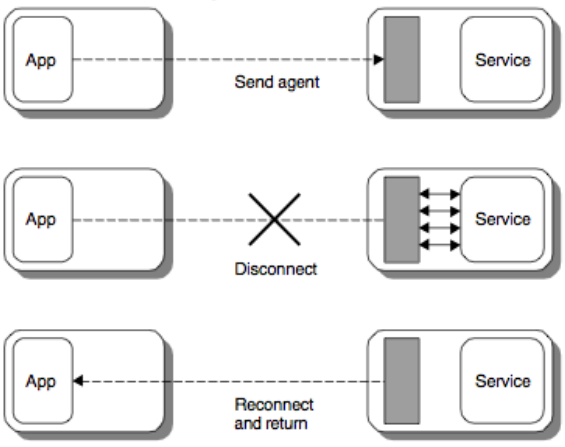


Figure 4.3: Asynchronous and autonomous agent execution [11]

Adaptability Agents sense their execution environment and are able to react on changes. Agents support the ability to distribute themselves to different platforms to maintain their optimal configuration to solve a particular problem [11].

Reduced Network Load Distributed objects rely heavily on communication protocols that involve multiple interactions between the "client" and the "server". This results in a lot of network traffic. Agents allow to packet the communication and dispatch it to a destination host where it gets processed. The further allow the definition of generic preprocessing agents, which reduces the raw data transmitted over the network [11], see also Figure 4.4.

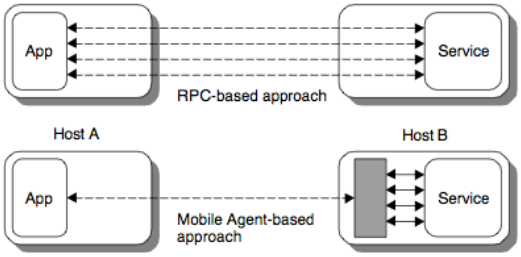


Figure 4.4: Agents reduce network load [11]

Fault tolerance and robustness Agents adaptability makes it easier to build fault-tolerant systems or define a secure default behaviour for unexpected situations. If a platform or instance gets shut down, all agents deployed on this instance will be warned and given time to either transmit their data to another agent or deploy themselves to another instance to continue operation [11].

For the SAaaS architecture, an evaluation of existing agent frameworks was carried out, with the following requirements:

- Dynamic, adaptive infrastructure
- Good performance
- Open Source software platform
- Documentation & community support

Since the CloudIA [27] architecture is built around the cloud management system OpenNebula [159], another requirement was a compatible agent programming language: Java. As a result, the Java Agent Development Platform (JADE) [160] was chosen, which enables the implementation of multi-agent systems. It consists of an active community and a frequently updated mailing list. Its agents are very lightweight. The learning curve of agent development was not steep, which could lead to a good market acceptance of the project. "JADE can be considered an agent middleware that implements an Agent Platform and a development framework. It deals with all those aspects that are not peculiar of the agent internals and that are independent of the applications, such as message transport, encoding and parsing, or agent life-cycle."[160]. Furthermore, it already provides a user interface, which alleviates agents creation, deployment and testing. The Java Agent Development Framework met all the established requirements, introduced above.

Java Agent Development Framework

The Java Agent Development Framework supplies the basis of the SAaaS' security and audit agents. It allows creation, deployment and orchestration of lightweight software agents. JADE agents are small, Java applications, which are developed to execute a specific task, dependent on a required entry behaviour. JADE is a multi agent system, which defines agents according to the IEEE Computer Society standards organisation for agent-based technology and its interoperability with other technologies (FIPA). It provides agent management components, such as Naming Service, Yellow- Page Service, Message Transport und Parsing Service. Agent communication is realised as messages, utilising the FIPA Agent Communication Language (ACL) [161] format. Main JADE characteristic is its implementation in JAVA, which results in a platform independent applicability. Each JADE platform runs at least one main container with

three default agents and contain more sub-containers. Default agents are an Agent Management System (AMS), a Directory Facility (DF) and a Remote GUI Agent (RMA), which provide agent management functionality and are described in more detail in *Chapter 7, Section 7.2*. Newly created agents are running in the JADE main container or a sub-container. Thus, each SAaaS enabled cloud component runs a JADE container, containing at least an AMS, DF and RMA agent [162].

Type of Software Agents

In the SAaaS architecture, mainly four different agent types exist: `Sensor agents`, `Audit agents`, `Metric agents` and `Management agents`. However, respecting location and task, two different agent classes were identified:

- Sensor agents
  - SAaaS sensor agents
  - SAaaS metric agents
  - SAaaS audit agents
- Management agents
  - SAaaS management agents

Sensor agents are agents, which get configured and deployed to a certain instance, for example a VM to execute a very specific task. They are configured in advance with a certain task (e.g. monitor a logfile, perform a list of audit checks) and distribute their results (events) to other agent(s). They run independently on a remote instance and are not necessarily connected to a central instance. In the SAaaS environment, SAaaS sensor agents, metric agents and audit agents fall into this category.

Management agents are agents, which serve an underlying supporting management task, such as: agent creation, agent deployment, event preprocessing and forwarding. Although they also run independently, by their assigned task, they are meant to interact with other agents to support them. In the SAaaS environment, SAaaS

management agents, such as the Event Aggregator agent fall into this category.

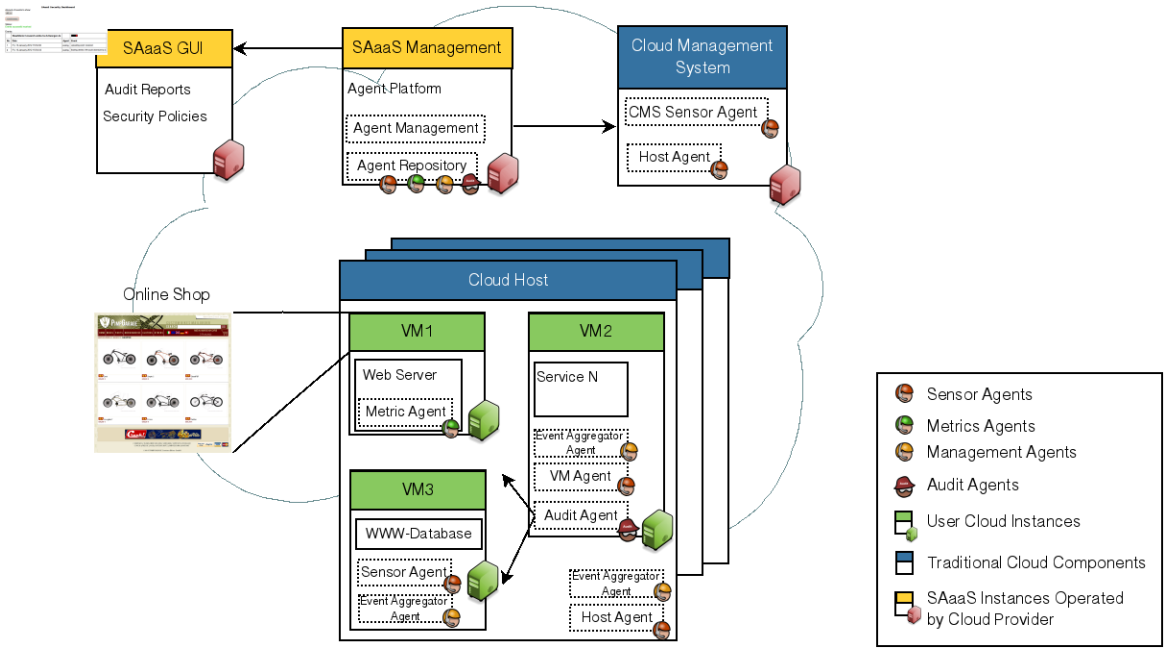


Figure 4.5: Types of SAaaS agents distributed in an IaaS Cloud

Figure 4.5, shows a high level IaaS cloud infrastructure with the different agents distributed over the different cloud components. Agents collecting data are called `Sensor Agents`. If the location of an agent needs to be expressed, they are also titled as `VM Agent` (agent running on a VM), `Host Agent` (agent running on a cloud host, monitoring for example hypervisor activities) or `CMS Agent` (agent monitoring the cloud management system). Specific targeted security audits perform specific checks of systems, which are affected by a change within the cloud environment. These checks are performed by `Audit Agents`. In case a specific, technical parameter needs to be validated to make a decision on a security status, this is called a threshold. Threshold values to be checked are defined as metrics and get checked by `Metric Agents`. To create, deploy and manage an agents life cycle and communication, `Management Agents` exist. For example, before an event from a specific sensor agent gets transmitted to another agent platform, it gets preprocessed and aggregated by an `Event Aggregator Agent`, which also runs on same location (agent platform) as the corre-

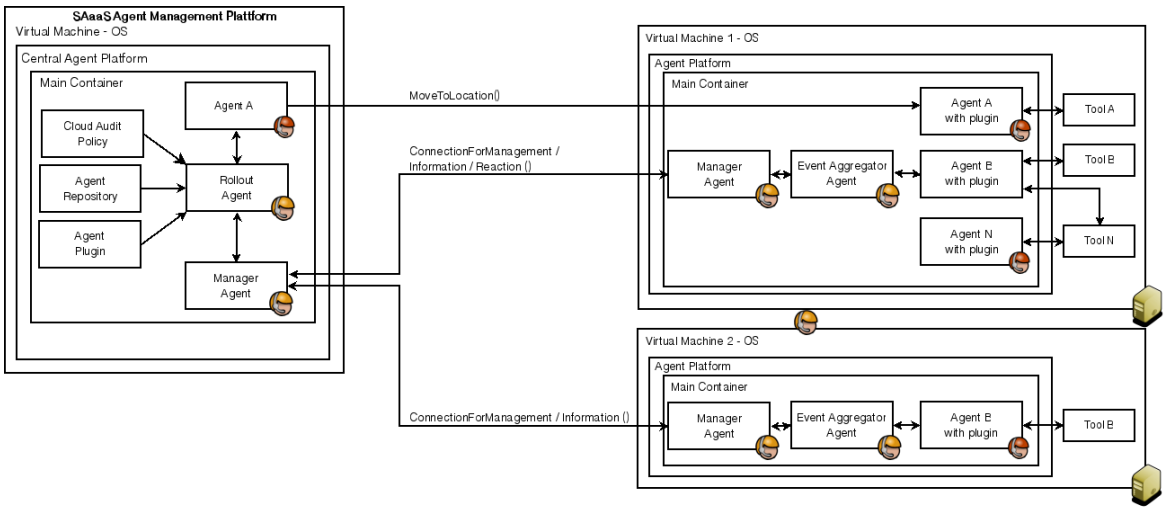


Figure 4.6: Basic SAaaS agent design

sponding sensor agent, e.g., a VM. Thus, the overall messages sent to a global cloud event processing system is reduced, which is especially important in large cloud computing environments. Details on specific agents are described in *Chapter 7, Section 7.2 - SAaaS Prototype Development*. A complete list of all developed agents, including technical details is provided in *Appendix A.2 - Developed SAaaS Agents*.

Agent Architecture

Figure 4.6 illustrates a basic SAaaS agent architecture. On the left side, it shows the SAaaS Agent Management Platform, deployed on a VM. It is responsible for a complete agent life-cycle: creation, configuration, deployment, monitoring, management and deletion. All agent life-cycle stages can be executed on-demand, which fulfils the requirement of a dynamic infrastructure. To create an agent, the SAaaS Agent Management Platform receives requirements from the Cloud Audit Policy module. This can include a configuration for a specific sensor agent (validate CPU load at VM x) or audit cases in case an audit agent is needed. Based on these requirements, an agent template is chosen and a new agent gets created, configured and equipped with additional tools, which are necessary for the agent. This could include a file mon-

itoring tool necessary to monitor a specific directory for changes at the target VM. As soon as the agent is build completely, it gets moved to the manager agent on the target agent platform on the target VM and starts working. Agents live in an agent platform, which provides them with basic services such as message delivery. A platform is composed of one or more containers. Containers can be executed on different hosts thus achieving a distributed platform. Each container can contain zero or more agents [162]. To provide monitoring or auditing functionality, a VM agents interacts through agent plugins with stand-alone tools, such as a process monitor, an IDS or an anti virus scanner. To harness the potential of cloud computing an agent can be deployed to a VM on-demand according to the policies a user defines. Inter-agent and inter-platform communication uses the Agent Communication Language Format (ACL). On the right side of Figure 4.6, two VMs with different deployed agents are depicted. Several tools are installed, feeding the different agents with information.

SAaaS Services in an IaaS Cloud

Figure 4.7 shows a more detailed view of SAaaS service components, which are available for every cloud user. The Figure shows instances of cloud "user 1", who runs two SAaaS enabled VMs in an IaaS Cloud. In addition to his two VMs, a SAaaS Agent Management Platform runs exclusively (for every user) on a separate VM. As described above, it is responsible for the agent management. Furthermore, it runs a Report Generator module, which conditions audit reports to a user readable format. Events, agent configurations, audit reports, user specific SAaaS configuration settings and behaviour data are stored in a user exclusive SAaaS database service. Additional SAaaS services are an Cloud Audit Policy Management and an Anomaly Detection module, which are described later in detail in *Chapter 5 - Cloud Audit Policy Language* and *Chapter 6 - Anomaly Detection* and are therefore greyed-out in

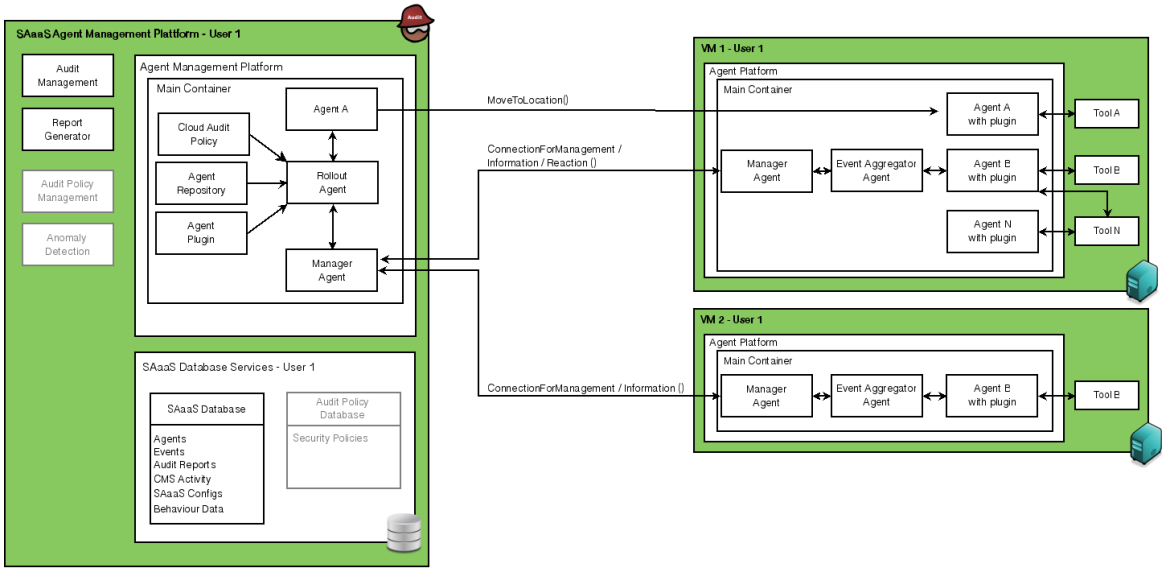


Figure 4.7: SAaaS Service Components - User View

Figure 4.7. Depending on the amount of SAaaS enabled cloud resources, the SAaaS management services are running on the same VM or can be spanned over multiple VMs. Thus, scalability is ensured. To achieve customer overspanning security, the same SAaaS components are deployed for the cloud provider, as shown in Figure 4.8. The formerly presented SAaaS user components (Figure 4.7) are combined into one box each "SAaaS User Management - User 1" - "SAaaS User Management User N". The provider's SAaaS Management Platform is depicted in orange. Despite investigating virtual instances of a specific user, the provider's SAaaS components are operating cloud wide. Thus, cloud wide security or audit policies can be deployed or a cloud wide incident or anomaly detection can be executed. For example, the provider's CMS agent stores all cloud management system actions, from all users in the provider's SAaaS data base, whereas the user specific SAaaS DB stores only CMS events of the corresponding user. Configuration of the SAaaS services is done through the SAaaS GUI. It is a web based interface allowing the user or provider to:

- Activate or deactivate SAaaS services for his VMs
- Manage agents (create, deploy, delete, move)

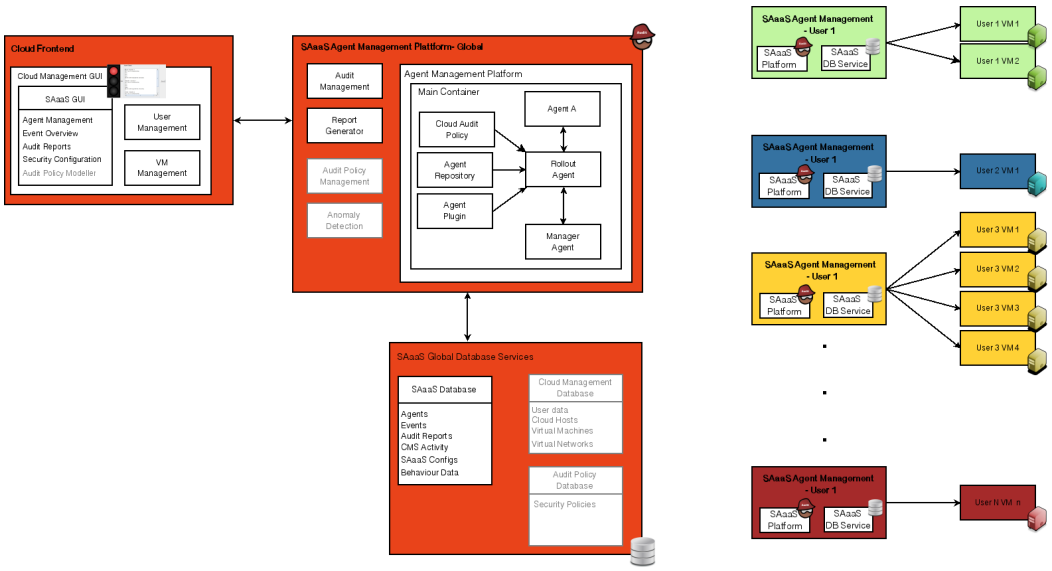


Figure 4.8: SAaaS Service Components - Provider View

- Investigate events (event log)
- See audit reports or perform manual audits
- Configure Security settings for his virtual instances
- Modell and manage (create, deploy, delete) audit policies

4.7 Evaluation: SAaaS Architecture

Since the developed Security Audit as a Service architecture is a major novel contribution of this research by providing an audit system which respects IaaS cloud characteristics, it gets evaluated in the following.

How Agents Can Improve Cloud Audits

Security incident detection in cloud environments is a non trivial task due to a cloud's characteristics. Using audit agents in an agent platform for cloud audits helps to commit to those characteristics, which can be described as:

Reduction of events Especially the frequently changing infrastructure poses a big challenge to define something, like normal behaviour and detect anomalous behaviour. Therefore it is important to have a high number of sensors capturing simple events. Simple events need to be preprocessed and abstracted to complex events, reducing the possibility "of event storms".

Fast adaption Combined with knowledge about business process flows it will be possible to detect security incidents in a frequently changing infrastructure while keeping the network load low. The usage of autonomous acting agents delivers this possibility.

Flexibility Agents can also be added, removed or reconfigurated during runtime without altering other components. Thus, the amount of monitoring entities (e.g., network connections of a VM, running processes, storage access, etc.) of a cloud instance can be changed without restarting the incident detection system. Simultaneously, using agents can save computing resources since the underlying business process flow can be taken into account.

Increased fault tolerance Using autonomous agents has advantages in case of a system failure. Agents can monitor the existence of co-located agents. If an agent stops for whatever reason this stays not undetected. Concepts of asymmetric cryptography or Trusted Platform Module technology can be used to guarantee the integrity of a (re-)started agent. However, this topic has not been addressed by this research work. If an agent stops, damage is restricted to this single agent or a small subset of agents, which are requiring information from this agent.

SAaaS agents are enabled to be aware of underlying business flows. A cloud provider can offer different business cases for VMs, such as "Online Shop" instances. Thus, preconfigured agents get automatically deployed with those instances. Within this architecture, several simple agents are logically belonging together forming an *agent group* where every agent knows about the other agent. While single agents monitor simple events (e.g., user login on a VM) they share them with other agents, as pre-

sented in *Chapter 7.5.2 - SAaaS Demo 3a - Login Bruteforce Detection.* Thus, cloud security audits get a wider spectrum of vision, not just focused on the security state of a single instance. Furthermore, by ordering agents in a hierarchical structure (multiple simple agents can exists on the same platform e.g., inside a VM), preprocessing of detected simple events and information sharing between logical associated agents leads to a reduction of network load [162]. Thus, the system is more scalable by reducing data sent to upper system layers. This is introduced and used in [163]. Combining events from sensor agents monitoring cloud hosts and VMs and infrastructure monitoring agents (network agent, firewall agent) incident detection is not limited to either host or network based sensors, which is especially important for the characteristics of cloud environments. This leads to an increased durability of the overall incident detection system. Furthermore, agents can be updated to new versions (depending their interface remains unchanged) without restarting the whole incident detection system or other SAaaS agents running at a VM.

Monitoring is done by simple sensor agents, like the introduced inotify agent. Analysis, planning and execution is done via the Event Aggregator agent. Due to the modular design this can be realised by several, single agents as well. Especially, the execution of actions dependent on the analysis result is implemented as an extra agent, thus can be used independently of the underlying event detection (e.g. file system change, infrastructure change, etc). One advantage, of this agent's intelligence is a reduction of manual user interaction. When deploying a new VM, dependend on the VM template a user has chosen, the SAaaS Agent Management knows, which agents need to be deployed to this VM. In case a new image gets used, first a software investigation agent could be deployed to the VM analysing, which software packages are installed on this particular VM. Dependend on the results, the SAaaS agent management knows, which agents should be deployed to this VM.

In the SAaaS enabled cloud a user has to define one or multiple VM Management

stations. These are basically IP devices, which are allowed to establish a management connection (e.g. SSH) to a VM. This prevents the SAaaS agents from sending false-positive events, like in the following scenario: Given a SAaaS enabled VM with an inotify agent already deployed monitoring file system changes. When the cloud user logs into this VM to change some configuration files, this gets immediately detected by the inotify agent. It sends these events to the local Event Aggregator agent. Instead of just forwarding the events to the SAaaS event management system, the Event Aggregator agent first checks, if an allowed management connection is established to this VM. Is this the case, the inotify events are classified as "allowed" not resulting in an alarm. Only a "configuration changed" event gets forwarded. When the management connection is closed again, the Event Aggregator Agent notifies the SAaaS Agent Management to automatically deploy a new Audit agent to validate the security state of this VM after this configuration change. This forms the concept of automatic concurrent cloud audits.

Agent Performance Measurements

It is essential for the SAaaS architecture, that the agents are very efficient not causing a high offset of resource consumption. Therefore, performance tests were executed to show the overhead introduced to the cloud VMs by the JADE platform. Additionally, it is tested, how fast agents can be deployed to a VM in runtime and second how fast the agent communication is. All tests are done at the university's research cloud infrastructure CloudIA. Table 4.1 shows the real and virtual hardware configuration of the utilised systems. Each test was performed at least 10 times, average values are presented in the following.

Performance Test 1 - JADE platform overhead

First, the introduced overhead by the JADE platform was tested. E.Cortese et al.

	Cloud Host	**Virtual Machine**
CPU	8x CPU: Intel(R) Xeon(R) CPU E5504 @ 2.00GHz 64-bit architecture	1 virtual CPU @ 2.00GHz 64-bit architecture
Memory	12 GB	512 MB RAM
Network	Gigabit Ethernet	Gigabit Ethernet
Storage	40 GB HDD SATA	4GB virtual HDD)
Storage protocol	SATA	NFS

Table 4.1: Evaluation: SAaaS agent performance lab setup

show in [164] that running the platform is not introducing a high CPU overhead. Additionally, it was measured how the boot process of a VM gets delayed because the JADE platform needs to be started. BootChart [165] was used to analyse overhead, see Figure A.3 - *Bootchart analysis of VM without SAaaS agent platform* and Figure A.4 - *Bootchart analysis of VM with SAaaS agent platform* in Appendix A.4. It can be confirmed, that in worse case the JADE platform adds two seconds to the total boot time of a VM of 13 seconds. For the assumed SAaaS usage scenario this is acceptable.

Performance Test 2 - Agent deploy time

Second, the time was tested it takes to create, configure and deploy an agent to a new platform. Therefore, five measure times are introduces:

- t0 - agent gets created
- t1 - agent gets started at the Agent Management VM
- t2 - agent loaded its configuration
- t3 - agent got transferred to target VM
- t4 - agent starts working on target VM

Figure 4.9 shows the involved JADE agents, developed for this test. The test was done first, using no connection security for transferring the agents (MTP HTTP). Second,

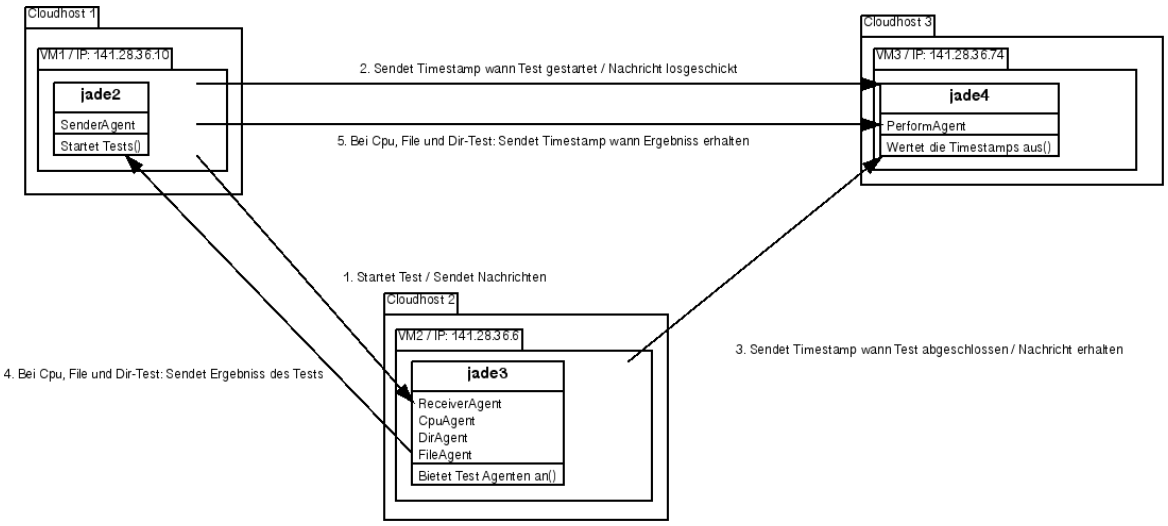


Figure 4.9: SAaaS Agent Platform performance test components: agent deploy time

a secure and authenticated channel between platforms was used (MTP HTTPS) to see how much impact secure communication introduces. Figure 4.10 confirms, that on average it only takes 180ms for a full deploy of an agent using MTP HTTP and 210 ms using MTP HTTPS. This proves the applicability of the JADE agent platform to support the presented SAaaS use case.

Performance Test 3 - Agent message time

Third, the agent message performance was tested. The roundtrip time of a variable amount of message with a variable payload (string length of event message) between two agents was measured. As a constraint a maximum roundtrip time of 1 second was set. Figure 4.11 shows, that with the JADE agents we are able to send more than 500 messages with a maximum payload of 10.000 characters at once before the limiting constraint is hit. This is acceptable for the introduced SAaaS scenario. However, the test also showed, that after sending 220 packets with a length of 1.000.000 Strings, a java.lang.OutOfMemoryError: Java heap space error was thrown at the receiver agent. This limit is due to a default configuration parameter of the Java VM and could be solved by adding the option “-Xms256m” to increase the heap size for this Java VM.

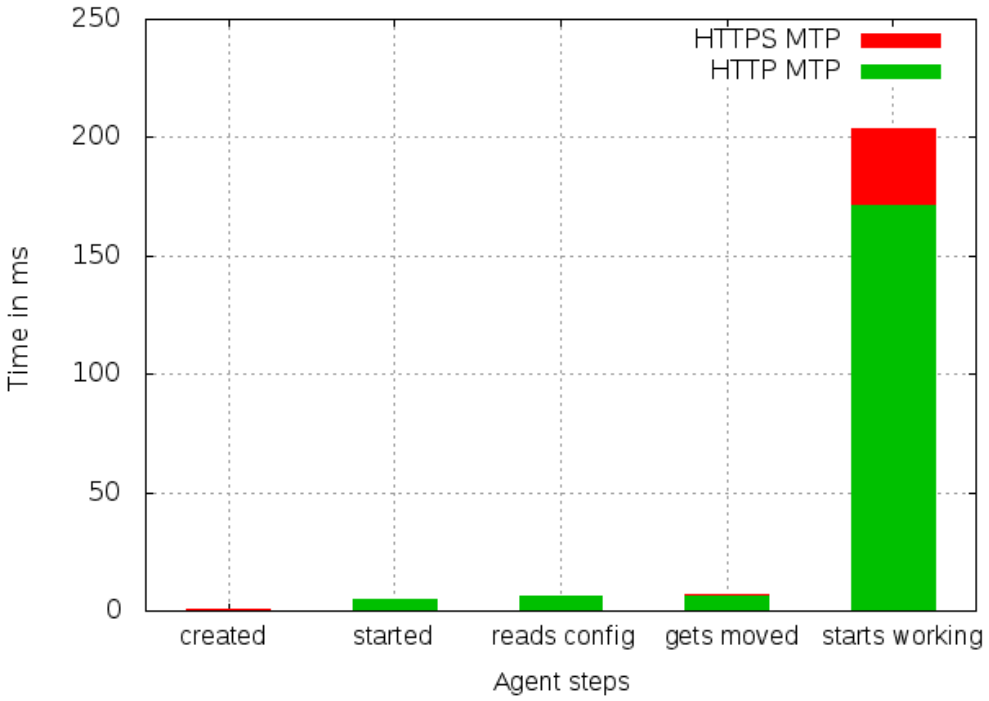


Figure 4.10: SAaaS Agent performance test: Agent deploy time

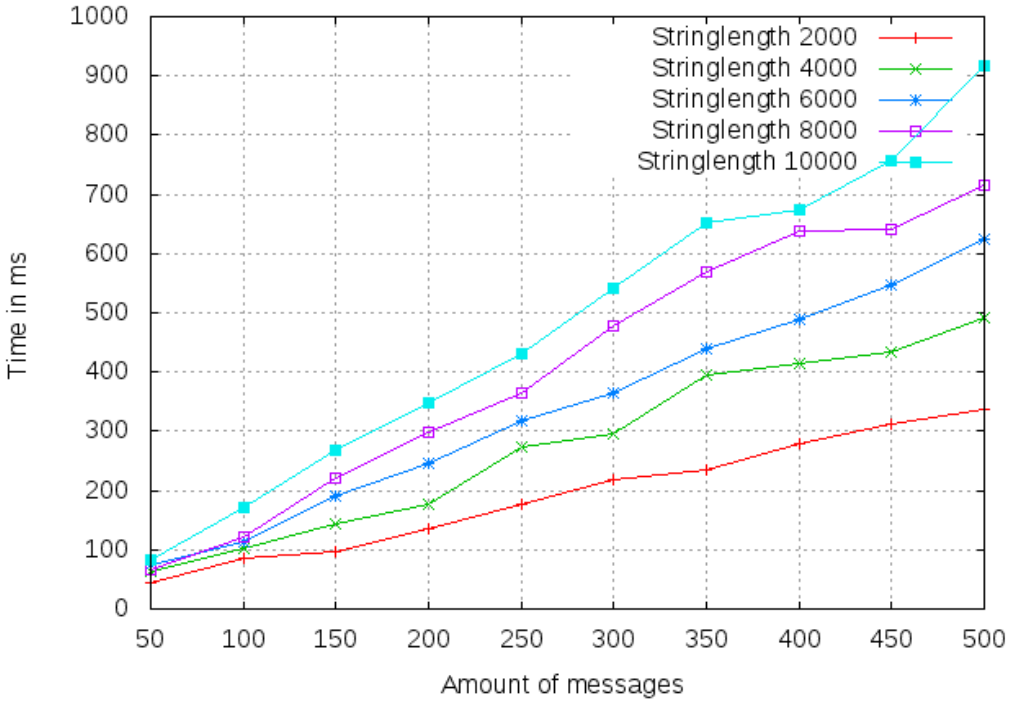


Figure 4.11: Agent message roundtrip time

4.8 Summary

Former research on cloud security issues (Chapter 3) identified that security audits could be a feasible approach to address the identified cloud security issues. The subsequent literature review on cloud audit systems showed, that their is a critical lack of cloud specific audit systems, respecting a cloud's characteristics. Thus, this research developed the Security Audit as a Service architecture to address the identified limitations. It is based on mobile, lightweight agents, which perform concurrent security audits in case a change within a cloud infrastructure is detected. Changes are detected by sensor agents, which are deployed at key points of a cloud infrastructure. Events are preprocessed and abstracted by management agents. In case a change got detected, audit agents are configured with very specific audits to evaluate the security status of the (change) affected cloud instances. An example use case concerning cloud scalability was presented and details on the utilised agent architecture discussed. A general agent design was elaborated. In a qualitative and technical evaluation it was shown that agents are a feasible approach to create a cloud audit system. The novel SAaaS architecture mitigates the following identified limitations in the area of cloud auditing:

- A technology independent, cloud provider interoperable audit system for IaaS clouds, which does not require special hardware
- A hypervisor technology independent sensor system utilising software agents

Besides those technical attributes, it provides more transparency in cloud infrastructures for cloud customers and providers. This enables cloud customers a better risk analysis of IaaS cloud infrastructures and helps cloud providers to maintain a secure cloud environment.

5 A Cloud Audit Policy Language

"Language is the blood of the soul into which thoughts run and out of which they grow."

(Oliver Wendell Holmes, Physician, poet, professor, author)

To be able to define security constraints and corresponding cloud audits, security policies need to be modelled. For implementation and especially for an automatic enforcement through concurrent cloud audits, a language representation of a concrete security policy is needed. Thus, the Cloud Audit Policy Language (CAPL) is developed. This chapter describes the requirements for a cloud audit policy language. Beneath generic attributes, six specific policy scenarios are introduced, which form the requirements for the target language. Afterwards, already existing security policy languages get introduced and evaluated. It is elaborated, why none of them fulfils the established requirements. Therefore, a new Cloud Audit Policy Language is introduced, which is based on the Cloud Infrastructure Management Interface specification. It forms the third novel contribution of the research, after the cloud audit test criteria catalogue and the SAaaS architecture.

5.1 Introduction

Cloud instances of a customer always serve a certain business case. Thus, different VMs require a different level of security protection, dependent on the service they are providing. The presented SAaaS agents are a technical tool to fulfil those different requirements in a flexible manner. However, they need to get an understanding how the desired nominal security state looks like, which checks (audits) need to be performed to validate the current state, and which tools are necessary for this task. Ideally, configuration needs to be done by the personal who knows the deployed instances best: the cloud provider's administration personnel for the cloud architecture, and the cloud customer for his cloud instances. Thus, it is essential for those groups to describe the security and privacy requirements, in a machine understandable way, which will enable automatic, concurrent security audits. This is commonly achieved by the definition of security policies, transferring a requirement into a checklist of one or multiple testable conditions. To respect cloud user's and provider's security requirements, both parties need to be able to create policies.

Summary of Research

This part of the research presents a novel cloud audit policy language, which is used to model security policies and cloud audits, which result in SAaaS agent configurations.

Parts of this research phase have been published in the following papers and presented to the research community at the corresponding conferences:

- Sun Behind Clouds - On Automatic Cloud Security Audits and a Cloud Audit Policy Language, *International Journal on Advances in Networks and Services 2013* [166]

Association within the SAaaS architecture

Figure 5.1 highlights, which part of the whole SAaaS architecture is addressed by the Cloud Audit Policy Language. It covers the Cloud Audit Policy Modeller, which consists of the modules:

- Policy Management - Definition and communication of security and audit policies
- Policy Server - Backend for the policy management
- Policy Modeller - Frontend for cloud user and provider to model policies

Since cloud audits are represented in the Cloud Audit Policy Language, the configuration settings of SAaaS Audit agents is dependent upon the security policy language as well. Thus, Agent Management is partly highlighted in Figure 5.1 as well.

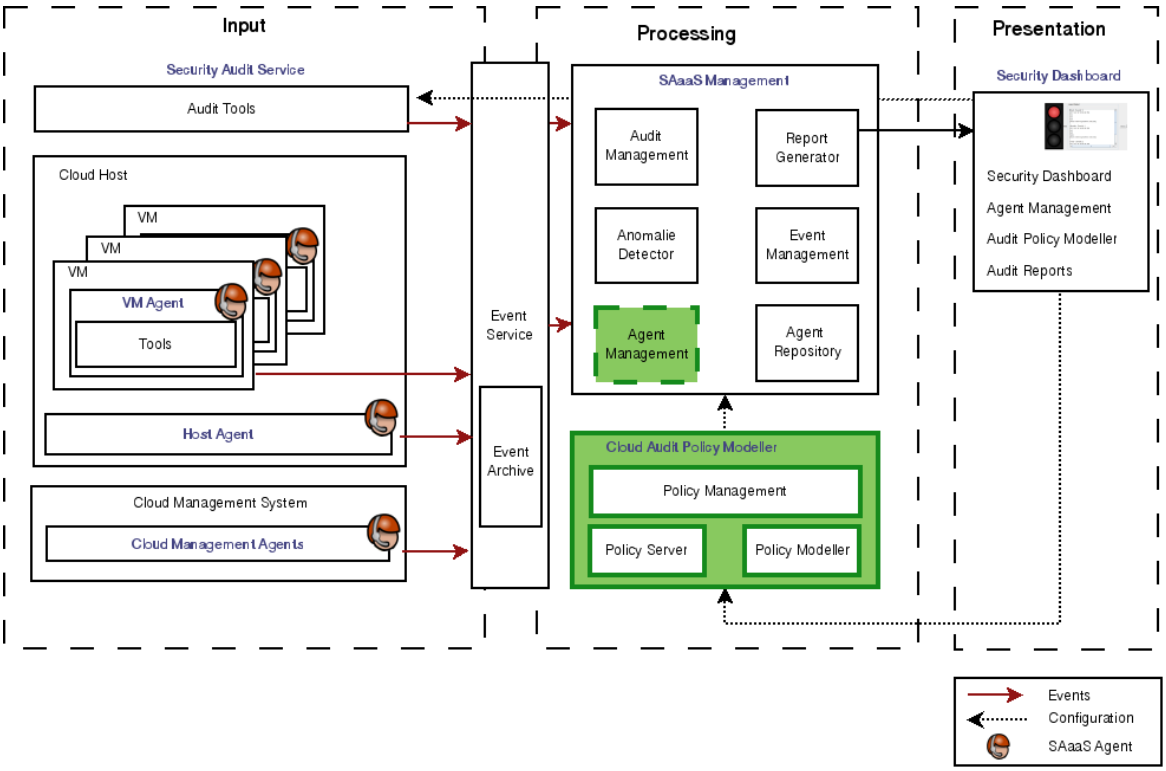


Figure 5.1: Affiliation of cloud Audit Policy Language within SAaaS architecture

Use Case Scenarios

Before analysing existing security policy languages, three use case scenarios are discussed, the target language should fulfil.

Modelling of security requirements of VM images

Public marketplaces for exchanging cloud appliances such as, OpenNebula Marketplace [114], Amazon Web Services EC2 Management Console or the Amazon Web Services Marketplace [115] provide cloud customers with an easy and efficient way of finding the right virtual machine image. But they also allow users to be administrators of their virtual machines, or upload and share their custom made VM images with other users. Although cloud providers provide security guidelines [116] on how to prepare an image before releasing it to a marketplace, current research by Balduzzi [91], Bugiel [117] and Meer [118] shows that marketplace images are highly insecure due to old software versions or "forgotten" or restorable security credentials, such as SSH private keys. Furthermore, a provider has interest in the VMs, running in his cloud environment as well. A single VM with a lot of security vulnerabilities could have an impact on the cloud's overall security state, since it could be used as a gateway for hackers to undercut a cloud providers security defence mechanisms. This could be due to an improperly configured VM or due to the usage of outdated software within a customer's VM. Thus frequent security audits of VM images are necessary, which could be executed every time before a new VM image gets used.

Modelling of technical and business process flow dependent actions

After deploying a VM to the cloud, a user needs to model his security requirements. This could be technical attributes, like allowed open network ports, or user access controls. Furthermore, the description of business process flows can improve security

in a highly dynamic infrastructure as a cloud as well. Consider an online shop scenario with web servers delivery database content in an IaaS cloud. A) If a request (using the allowed protocols) to the database VM without a preceding service request to the web application is detected this is rated as an abnormal behaviour, which does not occur in a valid business process flow. Therefore an event should be generated. B) If an upscale request for the user's web server instances gets registered at the cloud management system, without a corresponding high load on the web server VMs, this is considered abnormal as well. Business process flow dependent rules need to be modelled by a cloud user who is aware of its cloud instances and the underlying business process.

Automatic SAaaS agent configuration

After modelling security requirements and cloud audit checks, they need to be enforced. As a result, in the SAaaS system agents get configured with necessary tools (agent plugins, see Chapter 4, Figure 4.6) to monitor these requirements. Thus, a machine understandable security policy language is necessary, which allows the definition of a resulting SAaaS agent configuration. Since an automatic direct translation of all imaginable security policies into SAaaS agents configurations is considered unfeasible for a cloud provider, a community approach is estimated for a production use of this system. A cloud providers personnel, as well as cloud customer can develop modular agent configurations based on certain security policies. The network monitoring tool Nagios [167] is a popular example, where a huge user community provides Nagios plugins [168].

A key factor for the success of such a system is the detailed and distinct definition of security policies. However, this is contrary to a short VM deployment process a cloud user expects. Therefore, this work proposes to create a very easily operable, security policy generator, where cloud users can define security policies in a human way of

thinking, such as: "*The VM must be checked for malware*". Such simple policies could be supported by a graphical web interface with templates utilising check boxes or drop-down lists. This needs then to be translated into a machine understandable format, which results in the audit checks to be performed. The output of these checks needs to be translated back into a human understandable format, which will form the audit report submitted to the image creator, cloud provider and image user. To summarise, the target cloud audit policy language should provide the following possibilities:

- Modelling of technical security requirements
- Modelling of audit checks for VM images
  - VM images need to be audited in an automatic manner, to provide short response times to an image creator who wants to publish its image
  - The system needs to respect different security requirements from the image creator as well as the cloud provider
  - The system needs to produce a human understandable output in case an image did not pass the security check, providing the image creator with information about what prohibited the image release.
- Modelling of business flow dependent cloud actions
- Security policies need to be described in a machine understandable way

5.2 Related Work

Before analysing specific already security languages, related research work on security languages for clouds gets discussed. Literature review shows, that very little research in the area of cloud audit languages has been done. Morsy and Faheem identified the need for automated policy enforcement systems [169]. The main reason behind this argument is the increase in attacks and changes in organisational policies. Although,

a lot of different security policy definition languages exist (e.g., LaSCO, XACML, SPARQL, etc.), it is shown that each of those has different limitations in terms of policy constraints. Therefore, Morsy and Faheem propose a policy automation framework, including a new language called Standard Security Policy Language (SSPL), which tries to simplify the process of creating machine-readable security policies. The results of their policiy language analysis will be confirmed by the security policy language evaluation presented in this work. However, the presented Standard Security Policy Language seems that it only reached a research level, since to the best knowledge of the author, no public implementation or practical application using it could be found. During this research, the following additional security policy languages were investigated: Language for Security Constraints on Objects(LaSCO) [170], Rei Ontology Specifications (REI) [171], Ponder [172] & Ponder2 [173], eXtensible Access Control Markup Language (XACML) [174], SPARQL Protocol And RDF Query Language (SPARQL) [175], Open Vulnerability and Assessment Language (OVAL) [176] as well as business process languages like Business Process Execution Language (BPEL) [177], and Workflow and Agents Development Environment (WADE) [178] and the Intrusion Detection Message Exchange Format (IDMEF) [179]. They get discussed in more detail later in this Chapter, but before requirements for a cloud audit policy language need to be defined.

5.3 Requirements for a Cloud Audit Policy Language

Since the main target for a cloud audit policy language is to be able to transfer cloud security and audit rules into a machine understandable format two requirement classes on the language can be identified:

- Generic language requirements
- Specific language requirements: modelling of cloud audit policies

Generic Language Requirements

1. Technological Support Policies can be described, using textual as well as graphical methods. However, the focus of SAaaS will be on a textual description of policies. All language candidates will be analyzed with regards to their technological basis, especially whether they build upon established standards such as XML and JSON or they introduce completely new language formats. Using widely accepted technologies may be beneficial because there already exist a lot of tools such as parsers, interpreters and comprehensive documentation. Custom language formats however can be tailored to the problem domain and might improve flexibility and readability. To ensure a fast adoption by developers and leverage the large amount of tools already available, XML should be the preferred language base.

2. Development Activity Estimation Release cycles of tools, the size of the developing community and the adoption of a language by other projects indicates a high development activity and is an indicator for a future proof implementation.

3. Documentation Quality A comprehensive documentation is essential for understanding and evaluating a policy language. The quality of the documentation is hereby defined by factors such as the logical structure, accessibility, profoundness and consistency.

4. Complexity & Integratability in SAaaS The target language should be complex enough to fulfil all requirements but also generalisable up to a certain point. Too much complexity will affect the ease of learning by cloud administrators and therefore indirectly and negatively influence the utilisation of the language, which affects its overall success. Furthermore, it is essential to evaluate if the language can be integrated into the SAaaS architecture, presented in Chapter 4.

Specific Language Requirements

In addition to those generic requirements, there are specific language requirements to be considered when choosing a policy language. The most important criterion is the ability to model security requirements of cloud components, such as VMs and their interaction with each other. Beneath the generic requirements listed above, the target language needs to support:

- *Monitored Objects* Definition of any kind of entities in the cloud infrastructure, which shall be monitored (e.g., hosts, virtual machines, files)
- *Logical Policy Operations* Combination need to be supported, to create more complex policies by combining them with logical operators such as *AND* and *OR*
- *Policy Scoping* By grouping virtual machines or policies the process of creating and managing policies becomes easier. Also, incorporating the ability to define policies, which can only be used by the provider, may prove to be beneficial

To identify functional requirements for cloud audit policies, a bottom up approach was taken by defining the following example policy scenarios, which need to be describable by the security policy definition language:

No.	Policy Scenario	Example Policy
P1	**Malware** Since malware affects availability, integrity and confidentiality, every VM image needs to be checked for viruses and malware before being started within a cloud. Running VMs must be checked frequently.	*The VM is free of malware.*

No.	Policy Scenario	Example Policy
P2	**Monitoring of filesystem** Malicious attacks often result in change of the file systems' content, such as modification of config files or installation of malicious software. Therefore, a cloud audit policy should allow to define: a) A certain file (or folder) may not be changed at all. Every single change should raise an action. b) Validation of a certain file containing a specific content. Latter is most important in config files, which affect security configurations.	*File X has not been changed.*
P3	**Technical attribute modelling** Security is expressed if the infrastructure complies to certain technical attributes. A very simple rule defines the state of a network port (open/closed).	*Port 80 is open. Allowed network protocols: HTTP*
P4	**VM content** A VM contains software. Security can be increased by banning certain software products or limiting specific versions of a software, to prevent data leakage or just to be compliant to existing software licenses.	*Software X must (not) be installed.*
P5	**Process communication** By modelling, which process are allowed to exchange data, data leakage can be prevented and defence in depth measures can be applied.	*Program X is allowed to communicate with program Y.*

No.	Policy Scenario	Example Policy
P6	**VM scalability** One attribute of cloud environments is flexibility and on-demand availability of resources. Depending on a currently existing demand additional VMs can be added to a certain service cluster (VM upscale) and when demand lowers, VMs can be decommissioned again (VM downscale). But this could also be misused by attackers to compromise the availability of a customer's cloud based infrastructure, by downscaling VMs during a high demand period. Contrary, unnecessary upscaling of VMs increases the running costs of a customer.	*Upscaling of VMs in VM cluster "WWW-Server" is only permitted if average requests per second* $\geq N$*".*
P7	**Data traces** In case of VM marketplaces, users prepare VM images and offer them at the marketplace (role: appliance creator). It is important, that these images don't contain any personal information of the VM image creator, such as private key files or passwords, which could lead to a security breach. It is to validate that files are cleared e.g., history file, and critical information is securely wiped (and not be restorable anymore even with file carving tools).	*The VM does not contain any personal information.*

5.4 Evaluation of Existing Security Policy Languages

A huge number of policy definition languages exist (ASL, LaSCO, PDL, XACML, SPARQL, SSPL, OVAL, etc.), all aiming to model security policies. Furthermore, business process languages (BPEL, WADE, YAWL, ADEPT, etc.) are applicable as well, which increases the size of languages to choose from. After a first investigation upon the applicability of those languages, WS-Trust, IDMEF, SSPL, PAX PDL, CADF and KAoS were sorted out, since they have not been proven to be compatible for the SAaaS approach. For the SAaaS use case, five possible language candidates were identified for a detailed investigation based upon the requirements described above: REI, Ponder, LaSCO, CIM and CIMI. They are presented in more detail in the following subsection. However, in this work the focus lies upon whether the language is suitable for SAaaS[1].

REI

REI [171] is an OWL Lite based language, developed by Lalana Kagal in 2005. REI allows the definition of management, security, privacy and conversation policies [171]. These policies define the optimal behavior in a problem domain. A policy is hereby defined by the prohibition, permission or the obligation to perform an action on a target. It supports the definition of monitored object with "targets" and "users". Combination of multiple policies is supported by "Denotic Objects". Management of conflicting policies is solved through the definition of priorities. REI got its last update in 2005. Furthermore, there is no active user community available. Documentation is rare, a rough description of classes and some papers exist. Learning curve seems to be steep and long, since it is a complex and nested architecture. This is contrary to an easy entry aspired by the SAaaS architecture, since the user should be able to easily

[1] A detailed description and investigation of each language is given in [180].

Evaluation Property	Fulfilment	Description
Technological base	⊖	OWL Lite
Definition of monitored objects	✓	Via attribute *targets* or *user*
Combination of policies	✓	Policies can be created from more Denotic Objects (permitted/prohibited/mandatory)
Area of validity	✓	"Policy Domains" can be created through constraints and object groups
Conflict management	✓	Based on priorities
Last version	○	Last update 2005
Acceptance	⊖	Only in PhD thesis of author, no practical implementation
Community support	⊖	None
Documentation	⊖	Rough description of classes[171], paper, presentations
Complexity	⊖	Long training period expected, complex & nested architecture
Support of SAaaS policy scenarios	✓	Yes
Estimated implementation effort	○	Unclear
Integratability into SAaaS architecture	○	Advantages of OWL (semantics) not necessary
Miscellaneous	⊖	Although implementations are mentioned in literature, no source code or executables were found

Table 5.2: REI characteristics

create their own policies, and further more important developers should be able to easily provide the policy-agent-tool configuration. The focus on semantic technology is not needed for SAaaS and introduces needless complexity. Additionally, REI has no practical relevance nor has it spread beyond a PhD thesis, which it was developed for. These are knock-out criteria for applicability to the SAaaS architecture, since a wide acceptation is aspired. Table 5.2 summarises all REI characteristics according to the established requirements.

Ponder

Ponder [172] is a policy specification language, which already features tools and services for policy enforcement and evaluation. Ponder is based on a proprietary language format, Ponder Talk. One of the main concepts behind Ponder is the general-purpose object management system and message passing paradigm [173]. In this, the language

Evaluation Property	Fulfilment	Description
Technological base	⊖	PonderTalk (SmallTalk)
Definition of monitored objects	✓	Possible, through managed object, action and target syntax
Combination of policies	✓	Possible via an *Obligation policy*
Area of validity	✓	Concept of self-managed cell
Conflict management	✓	Supported
Last version	✓	Last update 2011
Acceptance	○	Multiple paper citations available
Community support	⊖	None
Documentation	⊖	Examples exist, but only for older Ponder version. Not for Ponder2
Complexity	○	Policy readability: human friendly, development is difficult
Support of SAaaS policy scenarios	✓	Yes
Estimated implementation effort	⊖	Very high, since Ponder includes its own agent framework
Integratability into SAaaS architecture	⊖	No, own agents need to be used. Different philosophy of policy evaluation
Miscellaneous		

Table 5.3: Ponder characteristics

is meant to be implemented in a way that the actual decision making process (deciding whether a policy is fulfilled or not) needs to be as close as possible implemented to its data source. In a cloud scenario this would mean, that the decision engine needs to be implemented on each single machine. This is a knock out criterion for the targeted SAaaS scenario, since this would highly increase complexity. Furthermore, there is no active community supporting Ponder and it requires the usage of own Ponder-specific agents. Therefore, the developed SAaaS agents won't be usable. Thus investigation into Ponder was not continued further. Table 5.3 summarises all Ponder characteristics according to the established requirements.

Language for Security Constraints on Objects (LaSCO)

The Language for Security Constraints on Objects (LaSCO) [170] follows a graph based approach to define policies. Allowed and forbidden actions can be modelled

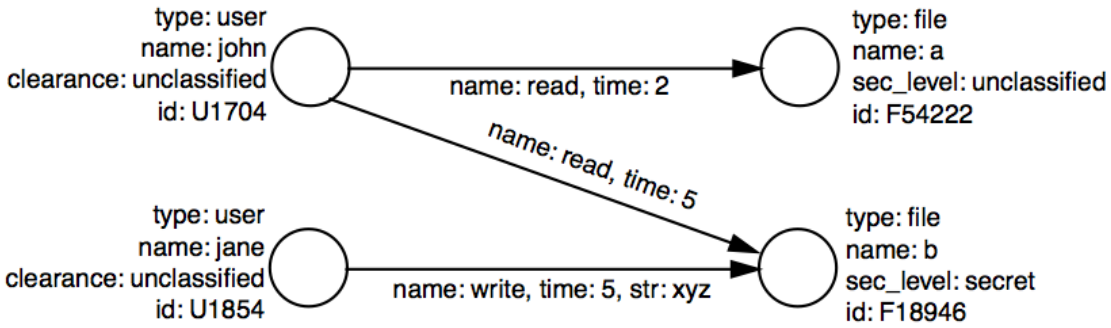


Figure 5.2: LaSCO example [12]

which get assigned to objects. Domains enable to define an area of validity in which a policy is valid. LaSCO comes with a Java based interpreter for graph interpretation. Figure 5.2 shows an example graph. "LaSCO Policies are annotated graphs that are matched to the system graph and which is then checked for adherence to the policy" [170]. Objects are represented as nodes and have several attributes, such as user or file, defined by the attribute *type*. Edges define events or actions. Following a graph based approach would introduces a lot of unnecessary complexity, since it would confront policy developers with a rather complex and steep learning curve. For the SAaaS system one requirement is it's simplicity and seamless support of commonly known technologies (see generic language requirements). Furthermore, no conflict management is supported by LaSCO, there is no community support and it has not spread over academic boundaries. Thus it is not suitable for the SAaaS scenario. Table 5.4 summarises all LaSCO characteristics according to the established requirements.

Common Information Model (CIM)

The Common Information Model (CIM) [181] is a model to describe elements and the relationships between them (such as network components or policies). Its specification gets developed by the Distributed Management Taskforce Inc. (DMTF) since 1997.

Evaluation Property	Fulfilment	Description
Technological base	⊖	Directed graphs
Definition of monitored objects	✓	Possible, via nodes
Combination of policies	✓	Possible via conjunctions
Area of validity	✓	Definition of Domains possible
Conflict management	⊖	Not supported
Last version	○	2000
Acceptance	⊖	Only PhD dissertation [12] and paper [170]
Community support	⊖	None
Documentation	⊖	Only examples in PhD dissertation
Complexity	⊖	Very complex due to graph based origin
Support of SAaaS policy scenarios	✓	Yes
Estimated implementation effort	⊖	Very high, completely different base
Integratability into SAaaS architecture	⊖	No, due to graph based nature
Miscellaneous		

Table 5.4: LaSCO characteristics

CIM is divided into the CIM Specification [181] and CIM Schema [182]. "The CIM Specification describes an object-oriented meta model based on the Unified Modelling Language (UML). This model includes expressions for common elements that must be clearly presented to management applications (e.g., classes, properties, methods, indications and associations)" [183]. The specification is the base for all CIM schemes. It defines the master schema, which contains structure und syntax of all CIM models. It is based on object oriented programming languages - familiar concepts are classes, inheritance or key attributes. Schemes contain collections of models and are hierarchically constructed. Each model is a collection of classes to represent data with its own meaning and area of validity, such as policies or networks. CIM provides currently 1400 classes for representing various elements and their connections. Models are written in UML and Managed Object Format (MOF). CIM was also extended with a CIM Policy model. It includes the following classes:

- PolicyCondition - Contains multiple sub-classes for representing dependencies, such as user, authentication, network packet filter, time periods.

- PolicyAction - One or multiple, reusable actions which can be assigned to one or more policies. They get executed, in case a policy condition gets into effect
- PolicyRule - The actual policy in the form of "if x, then y". Actions are executed if one or multiple conditions apply. "AND/OR" connections of conditions are possible.
- PolicyGroup - Allows grouping of single policies
- PolicySet - A collection of policies
- PolicyInSystem - Assignment of a policy to a certain object, such as a host.

During research, version 2.34 was investigated, current version is 2.38. It addresses most of the SAaaS requirements. Although CIM seems very compatible to the targeted SAaaS scenario it would require a lot of overhead added to the overall system. A complete CIM and WBEM implementation would be necessary, just to be able to define CIM policies. Since this would add an unnecessary amount of complexity and overhead to the SAaaS system, it has been considered not suitable for SAaaS.
Table 5.5 summarises all CIM characteristics according to the established requirements.

Cloud Infrastructure Management Interface (CIMI)

"The Cloud Infrastructure Management Interface CIMI)... describes the model and protocol for management interactions between a cloud Infrastructure as a Service provider and the consumers of an IaaS service. The basic resources of IaaS (machines, storage, and networks) are modelled with the goal of providing consumer management access to an implementation of IaaS and facilitating portability between cloud implementations" [186]. It is developed by the Cloud Management Initiative [187] which also belongs to the Distributed Management Taskforce Inc. Its main documentation is the CIMI specification [186] and the CIMI Primer [188]. CIMI provides a XML

Evaluation Property	Fulfilment	Description
Technological base	○	UML, MOF or XML (in combination with WBEM)
Definition of monitored objects	✓	Via association *PolicyInSystem* a policy can be bound to a VM
Combination of policies	✓	Multiple Conditions can be combined in a *PolicyRule*. Multiple policies can be combined to a *PolicySet*
Area of validity	○	Unclear
Conflict management	✓	Based on priorities
Last version	○	CIM 1998, policy model not clear
Acceptance	✓	Industry implementation exist: Windows Management Instrumentation [184], SBLIM project [185]
Community support	✓	None for policy model itself, but for specific implementations
Documentation	✓	Detailed documentation, but with inconsistency between different versions.
Complexity	⊖	High, due to huge CIM feature set and inconsistencies in documentation
Support of SAaaS policy scenarios	✓	Yes
Estimated implementation effort	⊖	High, since a complete CIM and WBEM implementation would be necessary
Integratability into SAaaS architecture	✓	Yes
Miscellaneous		

Table 5.5: CIM & CIM policy model characteristics (version 2.34)

based description language to manage cloud infrastructure components. Furthermore, actions can be modelled, such as starting and stopping of a VM or access management. Monitoring metrics for certain *aspects* can be defined, such as CPU load or network bandwidth. CIMI uses a REST based protocol for identification and access to modelled objects or actions. Thus, a management system can be implemented as web services, in a client server infrastructure. The server would manage and store objects described with the CIMI language. In the SAaaS context this would be cloud components and audit policies. The client(s) would create, request and edit objects through REST requests. In the SAaaS context, this would be cloud user creating policies or the agent management module requesting an agent-tool configuration. A CIMI described object, for example a cloud host with an ID 1, would be addressed by: `http://example.org/cimi/machines/1`. A CIMI XML schema of modelled cloud

objects is available under [189]. Objects can be described in XML or JSON. Listing 5.1 shows the creation of a new machine in CIMI, which could be either a cloud host or a VM. Line 1 and 2 show the request of a client, line 14 and 15 the server's answer. Line 4 - 12 contain information about an object (a new VM) to be created.

```
1  POST /machines HTTP/1.1
2  Content-Type: application/json
3
4 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineCreate",
5   "name": "myMachine1",
6   "description": "My very first machine",
7   "machineTemplate": {
8     "machineConfig": { "href": "http://example.com/configs/tiny"},
9     "machineImage": { "href": "http://example.com/images/WinXP-SP2"},
10    "credential": { "href": "http://example.com/creds/12345"}
11  }
12 }
13
14 HTTP/1.1 201 Created
15 Location: http://example.com/machines/843752
```

Listing 5.1: CIMI Example

CIMI was developed by a sub-organisation of the DMTF, thus it is oriented close to CIM. Although CIMI looks even more suitable than the previously presented CIM it does not include a policy model. This is a knockout criterion. However, since the language is almost supporting all established requirements, the to be developed Cloud Audit Policy Language presented in the next section will be based on CIMI. Table 5.6 summarises all CIMI characteristics according to the established requirements.

Evaluation

All language evaluation criteria are summarised in the evaluation summary, depicted in Table 5.7. All knock-out criteria (which do not fulfil the requirements introduced in Section 5.3) are displayed in bold red.

Evaluation Property	Fulfilment	Description
Technological base	✓	XML, JSON
Definition of monitored objects	✓	Description of IaaS cloud components exists
Combination of policies	✓	No policies, but combination of objects supported
Area of validity	✓	Tailored to IaaS cloud environments
Conflict management	⊖	Not existing
Last version	✓	Cloud Infrastructure Management Interface (CIMI) Model and REST Interface over HTTP Specification: October 2012
Acceptance	⊖	None
Community support	⊖	None
Documentation	✓	Detailed documentation, white paper, standard.
Complexity	✓	Tailored to IaaS cloud environments
Support of SAaaS policy scenarios	⊖	No policies phraseable
Estimated implementation effort	○	Moderate, policy extension needs to be developed
Integratability into SAaaS architecture	✓	Yes
Miscellaneous		

Table 5.6: CIMI characteristics (version 1.0.1)

REI is an OWL based language, which allows the definition of management, security, privacy and conversation policies. The focus on semantic technology is not needed for SAaaS and introduces needless complexity. Additionally, REI has no practical relevance nor has it spread beyond a PhD thesis, which it was developed for. CIM is a model to describe elements and the relationships between them (such as policies). It addresses most of the SAaaS requirements. Because of the policy model, CIM is suitable as a policy language for cloud audits. *PolicyActions* could be used to deploy corresponding agents in case a condition applies. However, an implementation according to the CIM standard would have gone way beyond the requirements of the SAaaS project. Implementation of a complete WBEM framework would have been necessary, just to support one CIM model (policy model). Ponder is based on a proprietary language (Ponder Talk), thus policy plugin developers would need to learn a complete new language. This is against the requirement of using a widely accepted technological base and does not promise to be future-proof. Additionally,

in the Ponder methodology, the decision engine needs to be implemented on each single monitoring instance. This is a knock-out criteria for the usage of Ponder for the SAaaS scenario. LaSCO follows a graph based approach. Although it might open up some interesting opportunities for a graphical policy modeller, the problem of conflict management is not addressed [170] and similarly to REI, LaSCO has not spread beyond academic boundaries (one dissertation in [12]). Furthermore, no active community exists. These points make this language unsuitable for SAaaS.

Existing security policy languages are either too complex, badly documented or very specialised for a specific context. It is shown, that none of the evaluated security languages fulfils the established requirements. As a result, an own cloud audit security policy language needed to be developed.

Evaluation Property	REI		Ponder		LaSCO		CIM		CIMI	
	Eval.	Details	Eval.	Details	Eval.	Details	Eval.	Details	Eval.	Details
Technological Base	⊖	OWL Lite	⊖	**PonderTalk (SmallTalk)**	⊖	Directed graphs	✓	MOF & UML, XML through WBEM	✓	XML, JSON
Definition of monitored objects	✓	Targets, user	✓	Managed Object, Action, Target Syntax	✓	Nodes	✓	PolicyInSystem	✓	IaaS cloud environments
Combination of policies	✓	Denotic objects	✓	Obligation policy	✓	Conjunctions	✓	Conditions, PolicyRule, PolicySet	✓	Combination of objects
Area of validity	✓	Constraints, objects groups	✓	Self managed cell	✓	Domains	○	Unclear	✓	IaaS clouds
Conflict management	✓	Priorities	✓	Yes	⊖	**Not implemented**	✓	Priorities	⊖	**Not existing**
Last version	○	Updated 2005	✓	2011	○	2000	✓	Currently revised (2013)	✓	2012
Acceptance	⊖	**Only PhD thesis, no practical application**	○	Cited in multiple papers	⊖	**Only PhD dissertation [12] and paper [170]**	✓	Windows Mgnt. Instrumentation [184], SBLIM [185]	⊖	None
Community support	⊖	**None**	⊖	**None**	⊖	**None**	✓	None for CIM, but for specific implementations	⊖	None
Documentation	○	Rough description of classes[171], paper, presentations	○	Good examples, but for old Ponder version	○	Only one PhD dissertation	✓	UML Diagram[182], Policy profile[190]	✓	Schema, Primer
Complexity	⊖	Long training period, complex & nested architecture	○	Policies are human friendly readable, but developing own is difficult	⊖	Very complex due to its graph based origin	⊖	High, due to inconsistencies of different versions	✓	Tailored to IaaS clouds
Support of SAaaS policy scenarios	✓	Yes	✓	Yes	✓	Yes	✓	Yes	⊖	**No policies, but tailored to IaaS clouds**
Implementation effort	○	Unclear	⊖	Very high, since it brings its own agents	⊖	**Completely different base layer**	⊖	**Complete CIM and WBEM implementation necessary**	○	Moderate, policy extension needed
Integratability in SAaaS architecture	○	**Semantic of OWL not necessary**	⊖	**No, own agents necessary, different philosophy of policy evaluation**	⊖	**No, due to graph based nature**	✓	Yes	✓	Yes

Table 5.7: Comparison of existing security policy languages

5.5 Cloud Audit Policy Language (CAPL)

From the evaluation of the existing security policy language, CIMI seems very suitable for supporting the SAaaS scenario requirements. However, it lacks a policy module. Thus, instead of developing a completely new language, the Cloud Audit Policy Language (CAPL) was developed as an extension of CIMI. CIMI already delivers a detailed object description of IaaS cloud components and a standardised protocol. Core features like the object model, the protocol and a simplified variant of CIMI classes (e.g., `Machine`, `MachineConfiguration`, `MachineImage`) are inherited by CAPL. However, due to the different focus of CIMI on managing cloud infrastructures some classes of CIMI have not been adopted in CAPL, because they are not required for the SAaaS scenario. Others were modified, extended or new classes for the support of policies were added. By staying close to the CIMI standard, it will be possible to define security policy for any CIMI compatible cloud infrastructure. Thus, the compatibility of the presented SAaaS system is increased.

CAPL User Roles

CAPL uses slightly simplified definitions of the CIMI roles *Cloud Provider* and *Cloud Consumer*. The cloud provider manages and provisions cloud services and possesses full access rights. The cloud consumer uses cloud services as well as the service for auditing his virtual machines. The cloud consumer has a limited set of access rights, which are required to define policies and triggering audits.

CAPL Service Interface

CIMI uses a REST based protocol for communication. CAPL adopts the CIMI service interface.

CAPL Language Base

The CAPL language format is technically oriented onto CIMI, which offers XML or JSON as a language format. An advantage of using a XML language is the availability of XML schema, which can be used for data validation. Thus, an incoming policy can be easily checked if all required arguments are provided. To be interoperable with other services and due to the wide distribution of XML in large distributed architectures, XML was preferred over JSON.

CAPL Definition

CAPL is based on a object/class model. Figure 5.3 shows the class diagram. As introduced, CAPL is designed to be oriented closely to the CIMI standard, but missing classes were added. CAPL enhances CIMI by adding several new classes:

- Machine

  The Machine class represents a machine, which shall be audited. CIMI uses Machines only for virtual machines. However, CAPL enhances the scope of Machines and includes host machines running virtual machines because those might be as well targets for audits.

- MachineTemplate

  The MachineTemplate defines the initial configuration of a VM.

- Policy

  Defines a policy rule (e.g., "a virtual machine must not contain malware"), which can be assigned to a machine or a group.

- PolicySet

  A PolicySet contains multiple Policies. Only if all contained conditions of the rules are fulfilled, the PolicySet evaluates to success. A PolicySet may be used

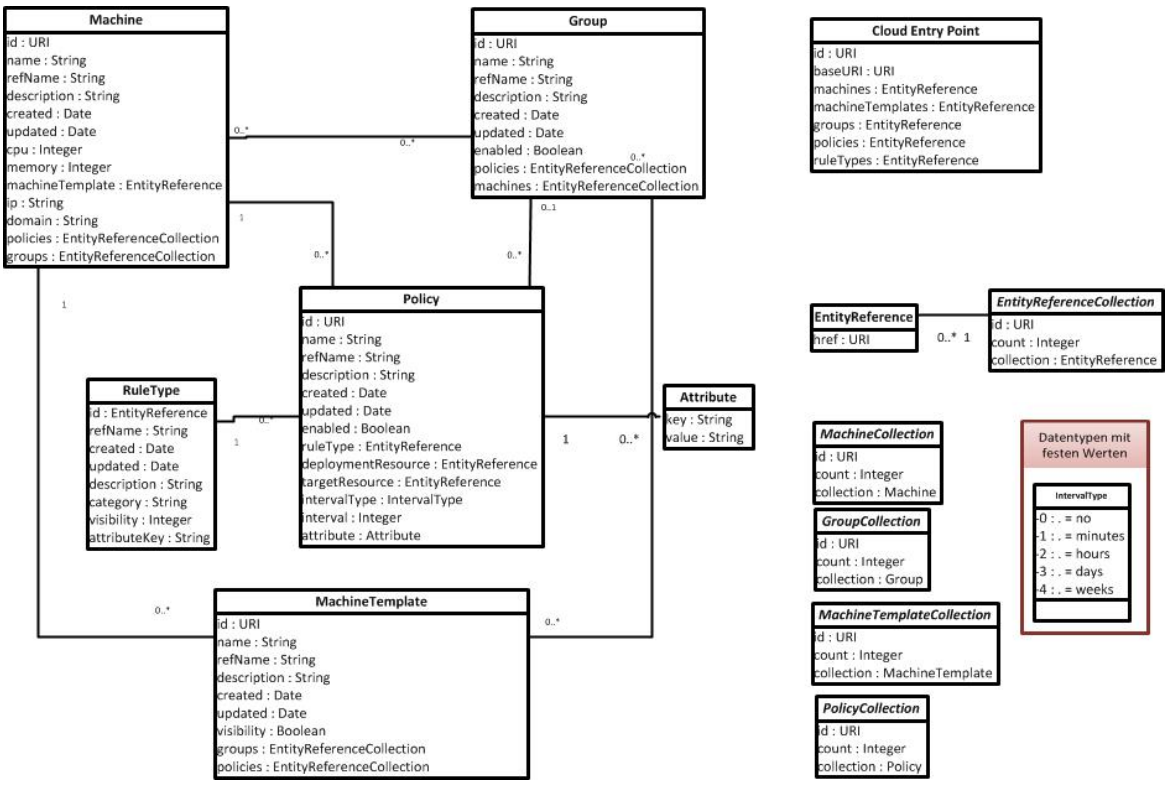


Figure 5.3: CAPL class diagram

like a policy and attached to machines or groups. Rules contained in a PolicySet may be linked disjunctive or conjunctive (using AND/OR). This behavior originates from the CIM policy model [191].

- Group

  Groups are used to manage related objects like multiple rules and machines. In such a case all rules of the group apply to all machines.

- RuleType

  RuleType describes what a policy is supposed to check and defines attributes and configurations, which the policy has to set.

In the following, the term "Policies" is uses as a interchangeably for the classes: `PolicyRule`, `PolicySet` and `MetricRule`.

CAPL Namespace

For CAPL, an own namespace has been assigned at `http://research.cloud.hs-furtwangen.de/capl/`. Thus, a distinct class identification is achieved and conflicts when using different schemes are avoided. Each CAPL class is assigned under the attribute `resourceURI` within the namespace.

CAPL Data Types

CAPL uses different data types. If possible, common data types, known from other programming languages were used. Table 5.8 lists available CAPL data types.

Data Type	**Name**	**Description**
Integer	Integer	A whole number
String	Text	A sequence of characters including text, numbers and special characters.
Date	Date	Timestamp
URI	URI	Contains a uniform resource identifier
EntityReference	Reference	A reference to an other object identified by a URI. Used within objects as `href=''uri''` and represented in the XML schema as `EntityReference`
EntityReference Collection	List of references	Gives back a list of references to class instances. Inherits from Abstract-Collection.
Attributes	Key/Value pairs	A list of self-defined attributes with a unique name and a corresponding value. Each key should be unique within a list. Represented as `<tagname key=keyname>value</tagname>`
Collection	List	A list of multiple data of one data type. Example: `EntityReferenceCollection` is a list of references. Each single data type also has a corresponding collection for creating an extensive list of this specific data type, such as `MachineCollection` or `GroupCollection`
IntervalType	Unit of interval	Describes the unit of an interval. Possible values are: • `no` - No Interval, one-time execution • `minutes` - value in minutes • `hours` - value in hours • `days` - value in days • `weeks` - value in weeks
ConditionType	Composition	Defines a composition of multiple rules. Possible values are either `D` - Disjunction or `C` for conjunction

Table 5.8: CAPL data types

CAPL Classes

For CAPL, the following classes are developed, also depicted in the class diagram, shown in Figure 5.3:

- CloudEntryPoint
- AbstractCollection
- Machine
- MachineCollection
- MachineTemplate
- Group
- GroupCollection
- RuleType
- RuleTypeCollection
- Policies
- PolicySet
- PolicyCollection
- Action
- ActionCollection

To support readability only the most important classes to understand the CAPL language specification are described, in the following. A complete class description of all developed classes is provided in Appendix A.3

Basic Class Structure

Each CAPL class consists of the following basic structure of attributes. They have the same meaning in every class, thus they are going to be presented here once and not included every time in the description of the corresponding classes. Table 5.9 provides a definition of the basic class attributes.

Data Type	**Name**	**Description**
resourceURI	URI	Contains the URI for identification of type within XML schema
Id	URI	Object ID under which it can be referenced
name	String	Name of instance
description	String	Description of corresponding instance
created	Date	Gets automatically set at object creation time
updated	Date	Gets automatically updated at object modification time

Table 5.9: CAPL classes: basic attribute structure

Cloud Entry Point

The Cloud Entry Point provides an interface for accessing the CAPL server. It provides an overview over all available resources and their corresponding URIs for a CAPL client (SAaaS Policy Modeller). It does not use the same basic structure, described in Table 5.9. For the SAaaS prototype (see Chapter 7) it is accessible at `https://cloud.hs- furtwangen.de/CAPLPrototyp/rest/`. Table 5.10 describes the attributes of the Cloud Entry Point.

Data Type	Name	Description
id	String	ID of Cloud Entry Point, e.g. `https://research.cloud.hs-furtwangen.de/ CAPLPrototyp/rest/CEP`
baseURI	URI	Contains the base URI, e.g. `https://research.cloud.hs-furtwangen.de/CAPLPrototyp/rest/`
machines	EntityReference	A reference for MachineCollection, which delivers all MachineTemplate instances, e.g. `https://research.cloud.hs-furtwangen.de/ CAPLPrototyp/rest/MachineTemplates`
groups	EntityReference	A reference for GroupCollection, which delivers all group instances, e.g. `https://research.cloud.hs-furtwangen.de/ CAPLPrototyp/rest/groups`
policies	EntityReference	A reference for PolicyCollection, which delivers all policy instances, e.g. `https://research.cloud.hs-furtwangen.de/ CAPLPrototyp/rest/policies`
ruleTypes	EntityReference	A reference to the list of all available RuleType, e.g. `https:// research.cloud.hs-furtwangen.de /CAPLPrototyp/rest/ruleTypes`

Table 5.10: CAPL class: cloud Entry Point

AbstractCollection

For every class, an additional class with the name schema `[Class-Name]Collection` is implemented, for providing lists and instances of this class. These "collection-classes" inherit from an abstract class `AbstractCollection`. Thus, each single class has the same structure as class AbstractCollection. No additional attributes exist. In case, a class contains a collection as an attribute, it is possible to get a list of instances or references (`EntityReferenceCollection`) for this specific class. Thus, only data which is necessary gets queried and submitted, saving unnecessary data overhead.

This design is inherited from the CIMI specification and applied to CAPL as well. Table 5.11 provides a definition of AbstractCollection.

Data Type	Name	Description
id	URI	ID of Collection under which it can be referenced
count	Integer	Number of objects in this list
collection	Collection specific data-type	A list of instances of this data type. Example: In case of `MachineCollection` it includes multiple instances of type `Machine`

Table 5.11: CAPL class: AbstractCollection

Machine

A `Machine` is an instance which needs to be audited. This can be a physical cloud host or a virtual machine. Audit `Policies` can be assigned to a `Machine`. `Machines` can be logically grouped together via the class `Group`. Machines are created by reading information from the cloud management system, which hosts and VMs exist. A definition of Machine is provided in Table 5.12.

Data Type	Name	Description
resourceURI	URI	Contains the base URI, e.g. `https://research.cloud.hs-furtwangen.de/CAPLPrototyp/rest/machines`
cpu	Integer	Number of CPUs
memory	Integer	Size of RAM
machineTemplate	EntityReference	A reference to MachineTemplate, which this instance is based on
ip	String	IP of instance
domain	String	Domain of instance
policies	EntityReferenceCollection	A list with references to policies which apply for this instance
groups	EntityReferenceCollection	A list of references to groups this instance is a member of

Table 5.12: CAPL class: Machine

MachineCollection

Delivers a list of instances of type `Machine`. Table 5.13 provides a definition of all attributes.

Data Type	Name	Description
id	URI	ID of Collection under which it can be referenced
count	Integer	Number of objects in this list
collection	Machine	A list of `Machine` instances

Table 5.13: CAPL class: MachineCollection

Group

The class `Group` provides the possibility to group `Policies` or `Machines`. All assigned policies of a certain `Group` apply for all `Machines` of this group.
Table 5.14 provides attributes of class `Group`.

Data Type	Name	Description
resourceURI	URI	Contains the base URI, e.g. `https://research.cloud.hs-furtwangen.de/CAPLPrototyp/rest/groups`
enabled	Boolean	Defines if group is enabled or not (not implemented in SAaaS prototype)
policies	EntityReference	A list of references on policies which are assigned to `Machines` of this group
machines	EntityReference	A list of `Machines`, which are member of this group. All member inherit all policies of this group

Table 5.14: CAPL class: Group

RuleType

`RuleType` defines the type of a rule and representing the SAaaS agent type. For each existing agent type a RuleType exists. RuleTypes are managed by the cloud provider. The class was defined to be very general, to support a variety of different rules. Since different rules contain different attributes, rule specific attributes get defined by the the attribute `properties`. Properties need to be provided at creation of a policy. They are used as a context-based attribute. They are stored in a hash map and assigned to the target agent upon agent configuration time. They are representing an agent's configuration. Context-based attributes can be assigned to simple data types, thus the CAPL server can evaluate if a context-based attribute is provided correctly and

can validate it. As an example a reference on a group can be assigned to the data type `GroupReference`. To provide information on available RuleTypes, a policy developer can access them from the CAPL server via a `GET` request. A definition of RuleTypes is given in Table 5.15. For the SAaaS prototype, RuleTypes for the defined policy scenarios presented in Section 5.3 are implemented.

Data Type	Name	Description
resourceURI	URI	Contains the base URI, e.g. `https://research.cloud.hs-furtwangen.de/CAPLPrototyp/rest/ruleTypes`
category	String	Enables categorisation of RuleTypes for sorting possibilities
visibility	Boolean	Defines if a RuleType is accessible for cloud user. Used for development, for beta RuleTypes during development
attributeKey	String	Context-based value, which needs to be defined in a policy.

Table 5.15: CAPL class: RuleTypes

Policies

Each policy contains the same basic structure, as shown in Table 5.16. Additionally, each policy contains policy specific attributes with context-based attributes. They are depending on the type of policy and defined by the attribute `RuleType`. They are resulting in an agent's configuration. `Policies` can be assigned to `Machines`, `MachineTemplates`, and `Groups`. A possible policy could be "Configuration of web server apache2 is finished". This results in a policy configuring a SAaaS `inotify agent` to monitor the file httpd.conf for filesystem changes. This functionality is implemented with the context-based attributes, defined by the attribute `attribute`. Its context is defined by the corresponding `RuleType`. Listing 5.2 elaborates how the attributes correspond to each other: The listing shows a policy which forbids changes on the file httpd.conf. Line 2-10 are common attributes, which are equally available in different policies. Line 11-15 show the context-based values, which are defined by the RuleType `inotify` (line 8). Line 12 states that a change (attribute action, line 13) of

the file httpd.conf (attribute `path`, line 14) is not allowed (attribute `permission`, line 12).

```
1 <PolicyRule>
2 <name>Freeze httpd.conf</name>
3     <refName>freezehttpd</refName>
4     <description>httpd.conf is final, must not be changed.</description>
5     <created>04.02.2013</created>
6     <updated>04.02.2013</updated>
7     <enabled>true</enabled>
8     <ruleType>inotify</ruleType>
9     <targetRessource href="http://example.org/machines/www1" />
10    <intervalType>no</intervalType>
11    <properties>
12  <permission>denied</permission>
13  <action>change</action>
14  <path>/etc/apache2/httpd.conf</path>
15    </properties>
16 </PolicyRule>
```

Listing 5.2: CAPL Policy: config freeze httpd.conf

Data Type	Name	Description
resourceURI	URI	Contains the base URI, e.g. `https://research.cloud.hs-furtwangen.de/CAPLPrototyp/rest/policies`
enabled	Boolean	Defines if policy is enabled or not
deploymentRessource	EntityReference	Is used in case a policy gets executed by a different `Machine`.
targetRessource	EntityReference	`Machine`, `MachineTemplate` or `Group` a policy is assigned to
intervalType	IntervalType	Defines interval a policy should be executed
interval	Integer	Defines value of interval
attribute	Attribute	Key/value pair containing context-based attributes of policy. Context is defined by `RuleType`. Can be used multiple times within a policy

Table 5.16: CAPL class: Policies

PolicySet

A `PolicySet` is a `Policy` which contains multiple policies. A PolicySet can be assigned to a `Machine`, a **MachineTemplate** and a `Group`. The attribute ConditionType

defines, how the policies are connected with each other. Possible values are: D - Disjunction, or C - Conjunction. In case of a conjunct connection, all policies need to be fulfilled for the PolicySet to be fulfilled. This corresponds to a traditional `AND` connection. In case of a disjunctive connection, only on policy needs to be fulfilled fort he PolicySet to be fulfilled. This method was adapted from the CIM Policy model [191]. A description of class PolicySet is provided in Table 5.17

Data Type	Name	Description
resourceURI	URI	Contains the base URI
enabled	Boolean	Defines if PolicySet is enabled or not
conditionType	ConditionType	Defines Type of Condition. For SAaaS prototype values `D` or `C` are available
targetRessource	EntityReference	`Machine`, `MachineTemplate` or `Group` a policy is assigned to
policies	EntityReferenceCollection	List of URIs to PolicyRule or MetricRule
intervalType	IntervalType	Defines interval a PolicySet should be executed
interval	Integer	Defines value of interval

Table 5.17: CAPL class: PolicySet

Action

`Action` enables the executed of operations. It is used for execution policies and if a `Machine` is added to a `Group`. Its definition is provided in Table 5.18.

Data Type	Name	Description
resourceURI	URI	Contains the base URI
action	String	Defines type of action. Possible values for SAaaS prototype: • **addMachine** - add a `Machine` to a `Group` • **removeMachine** - removes a `Machine` from a `Group` • **run** - start audit
target	Reference	Contains resource which the action applies to. In case of `addMachine` it contains the `Machine` which to add to a `Group`.

Table 5.18: CAPL class: Action

As already mentioned, a comprehensive definition of all CAPL classes is provided in Appendix A.3. Javadoc documentation can be found on the CD attached to this PhD

thesis in the folder "CloudAuditPolicyLanguageJavaDoc". CAPL is implemented as a client-server architecture using a database backend for serialisation of objects. Its technical implementation is described in Chapter 7, Section 7.4.

CAPL Library

All classes necessary for serialisation are stored in a CAPL library `libCAPL.jar`. It is used by the CAPL server and client for de-/ serialisation in XML or JSON. It enables the development of new CAPL clients. It is provided as Open Source software at the SAaaS website [139].
An example, which depicts the key features of CAPL, is shown in listing 5.3. This Policy describes the conditions under which upscaling of Web server VMs is allowed. In this case, it is measured whether upscaling is allowed or not.

5.6 CAPL Integration into SAaaS Architecture

The Cloud Audit Policy Language architecture consists of three major modules. A graphical user interface (Policy Modeller), the CAPL server and a database where policies are stored.

Policy Modeller is a graphical user interface of CAPL which is integrated into the SAaaS web GUI. It provides possibilities for policy definition and management. It communicates with the CAPL server over an XML protocol.

CAPL Server A REST service, which manages the cloud audit policies. It gets accessed over a standardised HTTP methods. It is responsible for creation, alteration and deletion of policies and answers policy requests. It forms a single instance for policy management.

CAPL Database The CAPL server stores all cloud audit policies in a relational database.

Policy Creation

Figure 5.4 shows the CAPL and SAaaS components in a simplified IaaS cloud architecture. As an example, creation of a scalability policy is assumed, with a following audit of this policy. In blue, the process of a policy creation is described. A user creates the following example policies ① via the graphical CAPL Policy Modeller (or directly through the REST interface):

a) Config freeze for configuration of web server

b) Scalability threshold: 100 requests / second

The policy modeller serialises the data in XML and send them via HTTP POST to the CAPL server ②. The policies get saved into the policy database ③. Since policy a) defines a monitoring of the web server's configuration file, an inotify agent gets created ⑤ and deployed ⑥ to the target VM, where it installs the necessary tool (inotify) and starts working ⑦.

Audit Execution

In case a user requests a manual audit ①, or an automatic concurrent audits gets issued 1a the `run` method at the CAPL server gets executed ②. In this example, it is assumed that an upscale event was registered at the cloud management system and necessary policies are loaded. The CAPL server fetches all applying policies via the `Machine URI` from the policy database ③ and checks, which agent is necessary for these policies. This information is forwarded to the SAaaS Agent Management ④, which creates the corresponding audit agent ⑤, and deploys it to the target VM(s) ⑥. When the agent arrives at the agent platform within the target VM(s), it starts

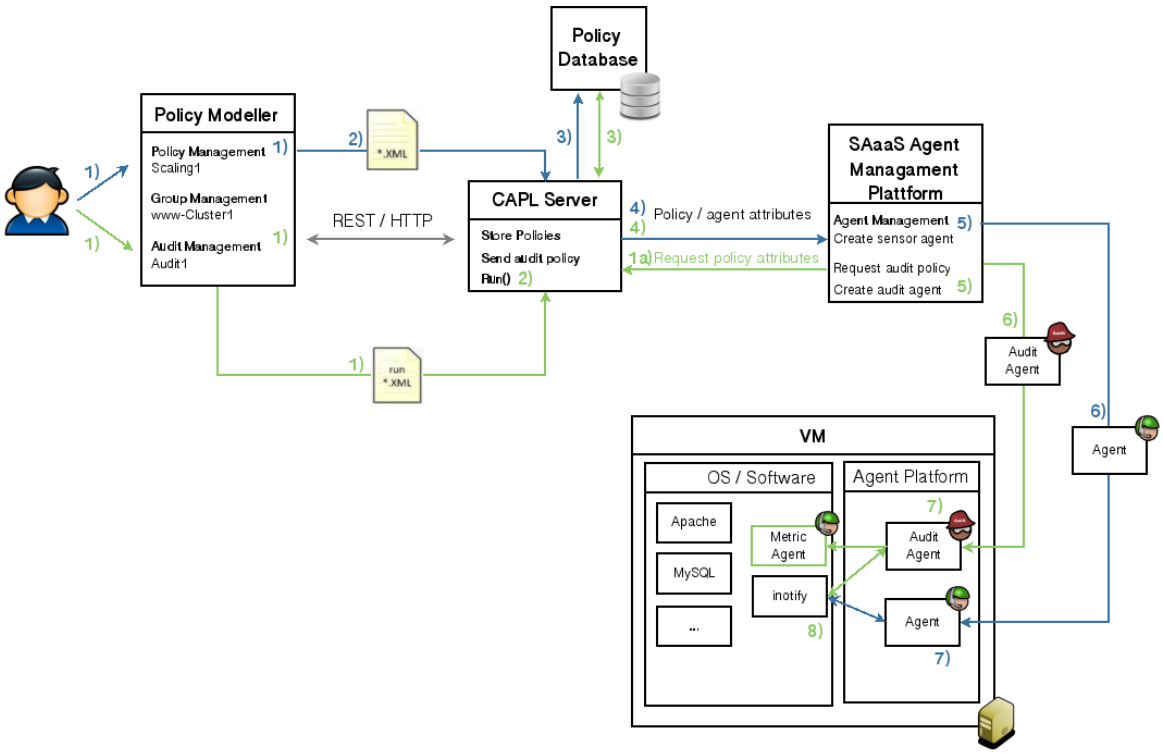


Figure 5.4: Integration of CAPL within SAaaS architecture

with executing its audit tasks ⑦. In this example, a metric agent checks the current http request count ⑧, which gets reported back to the Scalability Agent at the SAaaS Agent Management Platform.

Listing 5.3 shows the resulting policy in CAPL presentation. This policy describes the conditions under which upscaling of Web server VMs is allowed. Line 2 represents the `URI` of the web server group *WWWCluster1*. Line 11 and 12 state that two machines (ID 6 & 7) are members of this group. Line 17 shows the URI of the corresponding upscale policy, which has a context-based RuleType attribute `upscale` with a `metric` of *requests per second* and a corresponding `threshold` of *10*.

```
1 <Group xmnls="http://research.cloud.hs-furtwangen.de/capl/">
2 <id>https://research.cloud.hs-furtwangen.de/CAPLPrototyp/groups/wwwCluster1</
      id>
3 <name>Cluster at Loadbalancer1</name>
4 <refName>WWWCluster1</refName>
5 <description>Group of web servers at loadbalancer1
```

```
6 </description>
7 <created>2013-03-03</created>
8 <updated>2013-03-03</updated>
9 <enabled>true</enabled>
10 <machines>
11   <machine href="https://research.cloud.hs-furtwangen.de/CAPLPrototyp/rest/
       machines/6" />
12   <machine href="https://research.cloud.hs-furtwangen.de/CAPLPrototyp/rest/
       machines/7" />
13 </machines>
14 </Group>
15
16 <Policy xmnls="http://research.cloud.hs-furtwangen.de/capl/">
17 <id>https://research.cloud.hs-furtwangen.de/CAPLPrototyp/policies/upscale</id>
18 <name>Upscale is Allowed</name>
19 <refName>upscaleCluster1</refName>
20 <description>Upscale is only allowed when requests on machines is higher than
       10</description>
21 <created>2013-03-04</created>
22 <updated>2013-03-04</updated>
23 <enabled>true</enabled>
24 <ruleType href="https://research.cloud.hs-furtwangen.de/CAPLPrototyp/rest/
       ruleTypes/upscale"/>
25 <intervalType>no</intervalType>
26 <targetResource href="https://research.cloud.hs-furtwangen.de/CAPLPrototyp/
       rest/machines/5"/>
27 <attribute key="metric">requests per second</attribute>
28 <attribute key="threshold">10</attribute>
29 <attribute key="cluster">https://research.cloud.hs-furtwangen.de/CAPLPrototyp/
       groups/wwwCluster1
30 </attribute>
31 </Policy>
```

Listing 5.3: CAPL Example

This is a simplified scenario for showing the whole process how security and cloud audit policies defined by CAPL integrate into the SAaaS architecture. In a real world scenario, a load balancer would be running in front of a group of content delivering web servers. In case of a detected upscaling request, the current requests can be measured

directly at the load balancer. However, this can be done on-demand by a SAaaS agent as well, especially if the load balancer software is not compatible with an existing security monitoring application. For scenarios with an increased protection need, a cloud user does not necessarily trust the load balance, since it could be a malfunction, or hacked by an attacker. Then, by deploying agents to the web servers in addition to the load balance, they can either confirm the reported load by the load balancer or indicate a load balancer malfunction.

5.7 Evaluation: Cloud Audit Policy Language

To describe cloud audit and security policies, a Cloud Audit Policy Language was needed. Chapter 5.4 already evaluated existing security languages against the requirements, established for the SAaaS architecture. Thus, it is not done gain here. However, a comparison of the developed language against the requirements is still missing up to this point.

Because none of the evaluated languages fulfilled the requirements of SAaaS on a policy language such as missing conflict management o combination of policies (see Table 5.7 - Comparison of existing security policy languages), while retaining a reasonable complexity, CAPL was developed, which specifically addresses all those requirements. CAPL extends the Cloud Infrastructure Management Interface CIMI by a definition of security policies. It provides a textual and graphical interface for the description of security policies. The language format is XML based, which provides user friendly readability and support of widely accepted technologies, such as parsers, interpreters or documentation. It provides the ability to model monitored objects, such as VMs, cloud hosts or technical details. With its CIMI base it is close to a standard for cloud infrastructure description. Logical policy operations, such as *AND* or *OR* allow the creation of complex description of security states or a combination of different policies.

By grouping instances or policies it is possible to easily apply policies to a wide range of resources. All established use cases (*Section 5.3 - Specific Language Requirements*) are supported by CAPL. However, developing a new policy language also has some negative aspects. The CAPL language so far is used in the SAaaS project only, which leads to a rather poor acceptance. Also, besides the SAaaS project members, there is no community surrounding and developing this language. Table 5.19 evaluates CAPL against the requirements, established in Chapter 5.3. By remaining closely to the CIMI standard, it will be possible to define security policy for any CIMI compatible cloud infrastructure. This also increases the compatibility of the proposed SAaaS system. Another advantage of CAPL is its simplicity, since it is tailored to the SAaaS target scenario.

Requirement	**CAPL** Supp.	Details
Technological Base	⊕	XML
Definition of monitored objects	⊕	Tailored to problem domain
Combination of policies	⊕	Groups
Area of validity	⊕	CIMI compatible infrastructures
Conflict management	⊕	Included
Last version	○	Under active but internal development
Acceptance	⊖	Not spread beyond SAaaS
Community support	⊖	Only SAaaS
Documentation	○	CAPL documentation [180]
Complexity	⊕	Tailored to problem domain
Support of SAaaS policy scenarios	⊕	Full
Implementation effort	○	Own development
Integratability in SAaaS architecture	⊕	Fully integratable

Table 5.19: Evaluation of CAPL

5.8 Summary

This chapter presented an evaluation of existing security policy languages to support the requirements established for the SAaaS environment. Since none of them fulfilled those, a new Cloud Audit Policy Language (CAPL) was developed and presented. Details on the language design and specification were given and an example scenario was drawn, which explained the integration of the language into the SAaaS architecture.

6 Anomaly Detection

"If you are different from the rest of the flock, they bite you."

(Vincent O'Sullivan, writer, poet and critic)

This chapter presents an anomaly detection system for Infrastructure as a Service clouds as a feasibility study to show, how the SAaaS system can be extended to address more advanced and future cloud security issues. It is based on cloud customers' usage behaviour analysis. A machine learning approach utilising neural networks is used to analyse and learn the normal usage behaviour of cloud customers. This enables the detection of user specific as well as cloud wide anomalies, which could indicate a security incident. It increases transparency for cloud customers about the security of their cloud instances and supports cloud providers to detect misusage of their infrastructure.

6.1 Introduction

As elaborated in Chapter 3, IaaS clouds are inherently complex, large scale, frequently changing infrastructures, which bring traditional monitoring systems to their limits [121]. This is amplified by cloud characteristics, such as on-demand availability, elasticity of resources and multi-tenancy. But, not only industry is taking advantages of these characteristics. Security incidents at large cloud service providers, like the distribution of the Zeus botnet in Amazon Elastic Cloud [79] as well as academic research [91] show, that cyber criminals also make excessive use of IaaS cloud environments. Traditional IDS setups are built around a single monolithic entity, which is not adaptive enough to do data collection and processing in an efficient and meaningful way [121]. Therefore, this part of the research will demonstrate that parts of these misusage cases of cloud environments are detectable by using machine learning techniques. This is, because a change in the cloud usage behaviour of a customer or multiple customers often goes along with the abuse of cloud resources. The anomaly detection system presented in this work is located within a cloud, provided by the cloud provider, monitoring cloud usage of single cloud users, as well as user-overspanning cloud usage.

Summary of Research

This part of the research shows, that anomaly detection based on behaviour analysis is a valid approach to identify more advanced, so far unknown cloud security issues.

Parts of this research phase have been published in the following papers and presented to the research community at the corresponding conferences:

- Anomaly Detection in IaaS Clouds, *5th IEEE International Conference on Cloud Computing Technology and Science 2013* [192]

- Validating Cloud Infrastructure Changes by Cloud Audits, *Eighth IEEE World Congress on Services 2012* [144] - Winner of 2nd place at IEEE Services Cup 2012
- An agent based business aware incident detection system for cloud environments, *Journal of Cloud Computing: Advances, Systems and Applications 2012* [145]

Association within the SAaaS architecture

The anomaly detecting is an own module withing the overall SAaaS architecture (see Figure 4.1, Chapter 4) It uses information from the SAaaS sensor agents to base its decisions on.

6.1.1 Use Cases for Anomaly Detection in IaaS Clouds

To achieve an effective anomaly detection, Winter et al. claim [193] that it is crucial that the types of events to be detected are defined in advance and the system is designed and configured accordingly. Therefore, first an analysis was undertaken to determine, which cloud specific security threats can be detected by anomaly detection. To achieve a valid classification, identified cloud specific security risks (presented in Section 3.4) were analysed and classified according to their detectability. Two classes of anomaly detection methods for cloud security risks were identified:

- **Rule based anomaly detection**. Some risks are detectable by a rule based approach. This is always feasible, in case a certain threshold can be assigned to a specific value, either by the cloud user or the cloud provider. In case of the earlier presented scenario of a finished VM configuration, it results in a security policiy, stating: "Configuration of $VM_{ID=1}$ is finished, each change of a configuration file indicates a security incident". Rules can be easily applied and successfully audited utilising the presented SAaaS agent framework.

- **Behavioural based anomaly detection**. Other cloud specific risks, are not easily detectable, since no clear rule can be applied to a define a normal behaviour. For example, if a user's cloud credentials got stolen and VMs are created on his account. From the providers point of view, the user is able to access their cloud management interface with his correct login credentials and only legal cloud actions are executed. No hacking or exploiting of vulnerabilities took place. However, this can still be a potential threat to the cloud infrastructure, since it could be misused to carry out follow-on attacks, such as hosting of a botnet [79].

Table 6.1 provides an overview of security risks, their corresponding security impact and their detection class. The main priority of the selected risks is on cloud specific risks for IaaS clouds. Some risks can be classified for both detection categories, dependent on the actual attack, which originates from this risk.
Before showing how anomaly detection on cloud security issues can be done, related work gets presented first.

6.2 Related Work

To mitigate specific security issues in cloud environments, this chapter elaborates an anomaly detection system. Consequently, related research work on anomaly detection in cloud computing environments is presented. First, other anomaly detection systems are introduced, followed by work on neural networks and their applicability for increasing security in cloud environments.

Anomaly Detection Systems

A good introduction to anomaly detection is given by Banerjee et al. in [194]. Starting with anomaly detection in general and its applications, they later go into detail about

Cloud Risk	Risk	Impact	Detection Type
Abuse of cloud resources	Unsecured instances	Instances of different users are used to distribute malware	Rule based
	Misuse of cloud instances	Compromised cloud instances are misused to execute follow-on attacks	Rule & Behavioural based
	Account misusage	cloud instances can be controlled by attacker	Rule & Behavioural based
Missing monitoring of cloud scalability	Cloud scalability attacks	Malicious upscaling causes costs for cloud user, malicious downscaling affects service availability of cloud instances	Rule based
Missing security monitoring	Cloud Login Bruteforce	IP range of cloud instances gets tested for weak authentication with one specific login credential combination	Rule based
	Attacks on cloud Services	cloud management system gets misued to execute malicious commands	Rule & Behavioural based
	Cloud account hijacking	Malicious creation, altering or deletion of cloud instances	Behaviour based
Shared Technology Issues	Virtualization Vulnerabilities	Vulnerabilities in Hypervisor enable access to the entire cloud infrastructure	Rule based

Table 6.1: Considered cloud security risks

various algorithms. The work is finished with a case study of anomaly detection in network intrusion detection, where the authors present a system in use at University of Minnesota, which in the past successfully detected various network anomalies undetected by SNORT, such as new worms or very slow network scans. Lazarevic et al. [195], García-Teodoro et al. [196] and Patcha and Park [197] provide overviews over different anomaly detection techniques for intrusion detection. They all discuss advantages and drawbacks of various approaches, each focusing on a different subset. Lazarevic et al. compare different outlier detection techniques using distance-based methods like local outlier factor, nearest neighbour of unsupervised support vector machines. The algorithms are benchmarked, using the 1998 DARPA Intrusion Detection Evaluation Data [198], where the former scores the best results.

Nascimento and Correia [199] describe a system for finding anomalies in HTTP requests which can also be applied to HTTP services running as SaaS. They compare

different algorithms of which most are based upon analysing string tuples of different lengths.

Zhang et al. introduce in [200] an anomaly detection system for cloud networks based on NetFlow data. The paper gives just a high level overview of a proposed anomaly detection system, rather than providing details on the detection algorithm. However, Netflow information can be used by the presented SAaaS anomaly detection system as an additional input classifier to enhance the anomaly detection.

Pannu et al. present in [201] that anomaly detection based on support vector machines is a valid approach for validating cloud dependability assurance. In contrast to the anomaly detection system presented in this work, unlabelled monitoring data are used, which are processed by a support vector machines algorithm. This might be interesting for the presented SAaaS anomaly detection system as well, since unlabelled data can be used. A performance comparison on the anomaly detection rate would be necessary to show which approach performs better.

Fu introduces in [202] a performance metric system for autonomic anomaly detection in cloud environments. It is based on decision trees to identify the most important performance metrics out of a broad spectrum. Although the authors state, that this is used in an anomaly detection system, none is presented. However, the proposed metric evaluation could be a valuable addition for the proposed anomaly detection system (presented in this work) to automatically evaluate more valuable input classifiers.

Neural Networks in Clouds

Artificial neural networks (ANN) have broad applications to real world business problems and have been successfully applied in many industries [203]. Maithili et al. use neural networks in a cloud environment to estimate the possible recurrence rate of tumours by using data from lymphatic node positive patients. They show a success rate of 95%. A cloud based development framework enhances the efficiency and ac-

curacy of diagnosis.Mahmood et al. propose in [204] an intrusion detection system in cloud computing environments using neural networks. Cloud traffic data contains a lot of ineffective information, which needs to be identified and removed. This process is commonly known as feature reduction as a preprocessing step to overcome the curse of dimensionality [204]. Neural networks are used to obtain the optimal number of features to build an efficient model for the intrusion detection system. Joshi et al. argue in [205], that cloud environments are a profitable target for DDoS attacks. To mitigate the danger, they present a cloud trace back model, which is built on neural networks to trace back the source of a distributed denial of service attacks. They show, that neural networks are able to successfully identify an attack origin by 91%. Although research starts using neural networks to mitigate issues in cloud computing, so far their potential has not been used to identify security issues or cloud specific security attacks on cloud environments. Especially based on an IaaS cloud's nature of a frequently changing infrastructure, there is huge potential to benefit from neural networks for security attack detection, mainly due to their ability to identify anomalies in a big parameter space. With this addition knowledge a cloud audit system can be enhanced tremendously. Another major obstacle in the development of cloud auditing systems is the lack of quantitative information upon attack data. So far, no free available cloud usage data set is available, which could be used for evaluating and tuning developed audit systems. The authors of [205] also suffer from the absence of a real cloud data set. While Joshi et al. avoid this problem by using the DARPA data set [198], this was not feasible for this research, since it does not contain any cloud specific data, such as cloud management events. Thus high effort is needed to either get this data or develop a data simulator to advance the area of cloud attack detection.

In the following, rule based and behavioural based cloud security risks get elaborated in more detail. Each of them can be divided into two subcategories:

- **User-centric anomaly detection**. Usage behaviour of a single cloud user is analyzed. Thus, attacks on the user's instances, misusage of the user's account or errors in billing and accounting could be identified. A cloud user is interested in this information, since it affects the security of his cloud instances.
- **Cloud wide anomaly detection**. Behaviour of all cloud customers of a cloud infrastructure is analyzed. The provider is interested in this information, since it is able to indicate misusage of a cloud infrastructure. The user may be a benefactor as well as the security of his cloud instances is affected.

Instead of opening up a third category for risks, which apply to both parties (provider and user), those risks get listed under the category of the party who has more value of the information.

6.3 Rule Based Anomaly Detection

Rule based risks are detectable by defining security policies for a certain security state, and a threshold of an allowed deviation from that state. If a monitoring system detects a certain event, it compares the current situation against the security policies.

6.3.1 User-centric Anomaly Detection Use Cases

Account misusage is a well known problem from the area of web application and thus, also affects the cloud's management web interface. An attacker can gain access to a victim's account by different means like hacking weak user credentials or exploiting security flaws present in the application [206]. In cloud computing however, access to an account does not only grant access to the victim's data but also to the data hosted on services, which run under the victim's account and thereby potentially to data of many service users. Additionally, the attacker could spawn multiple new cloud

instances, using them for his own malicious intent and thus financially damaging the victim as he has to pay for the instances. By bringing down the victim's services an attacker could potentially cause even more damage. The anomaly of misusage can be detected by the fact, that the customer usually uses his cloud resources mostly between certain times each day, for example 8AM and 5PM. This can be stated in a rule, provided by the customer. But suddenly, heavy usage is registered during the night times.

One more cloud specific security risk are cloud scalability attacks. If an attacker is able to execute VM upscaling commands, this results in increasing costs for the cloud user. Conversely, an attacker executes downscaling commands, this results in an decreased availability of a user's cloud instances. Further consequences are possible, such as loss of service transactions during a high demand period. Detection can be achieved with a security policy regulating, that each CMS scalability command should precede a demand verification to validate if a scaling event is valid. Demand verification could be done at multiple points, such as a load balancer or at the affected cloud instances, such as members of a web server cluster itself.

6.3.2 Cloud Wide Anomaly Detection Use Cases

The cloud's computing power can also be used to carry out attacks on other targets. One possibility is to aggregate many VMs and use them to perform a Distributed Denial of Service (DDoS) attack on a single target, thereby preventing others to use its services. To detect such attacks, the network has to be monitored for abnormal activity especially from the inside. Due to the distributed nature of cloud computing information about network flow has to be collected at many different physical locations. To get the whole picture however, this data has to be analysed in the overall context. An example scenario is depicted in Figure 6.1: An attacker creates only a few number

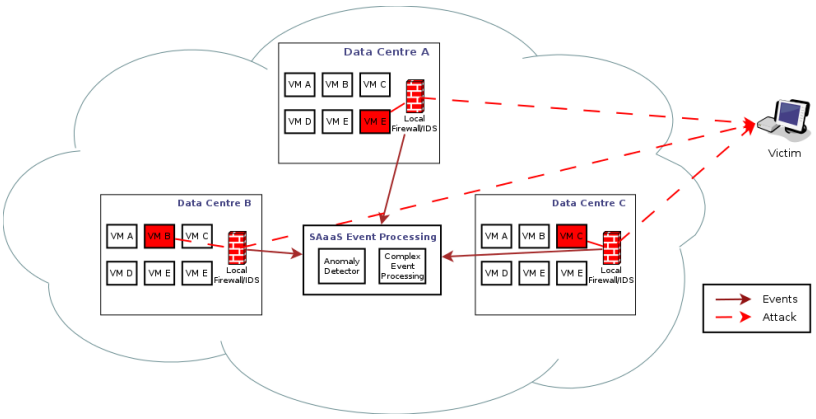


Figure 6.1: DDoS Attack Scenario

of instances in each of the cloud's data centers. Then he uses all of these distributed instances to DDoS a victim's host. If only the data in each data center was analysed for its own, maybe no attack would be detected as only a few hosts take part in it. If however, the data is combined and analysed in an overspanning detection system, this attack can be perceived in its whole extent.

A cloud infrastructure can run a huge number of systems, maintained by different cloud users, resulting in a heterogeneous level of security configuration. Mostly these instances are close together within a limited IP range. This is very attractive for misusage. Traditional bruteforce attacks on logins, such as SSH, MySQL, etc., try numerous combinations of username:password couples on one single target IP. Detection is simple, since this behaviour results in massive "login denied" messages in the SSH server's logfile. Therefore, sophisticated cloud login bruteforce attacks perform a "slow attack", by trying only one specific user + password combination on a whole IP range of a cloud provider's instances, instead of flooding one victim with millions of possible login credentials from a dictionary [207]. Consequently, the attack is hard to detect for traditional monitoring systems, like an IDS. By monitoring login attempts cloud wide over multiple cloud customer instances, a successful login with (multiple) preceding unsuccessful login attempts at different cloud instances can be defined as

abnormal behaviour and thus detected. Figure 6.2 depicts a cloud login bruteforce attack on SSH. This risk is not only limited to login attacks, but applies also for information gathering attacks, such as a slow port scan.

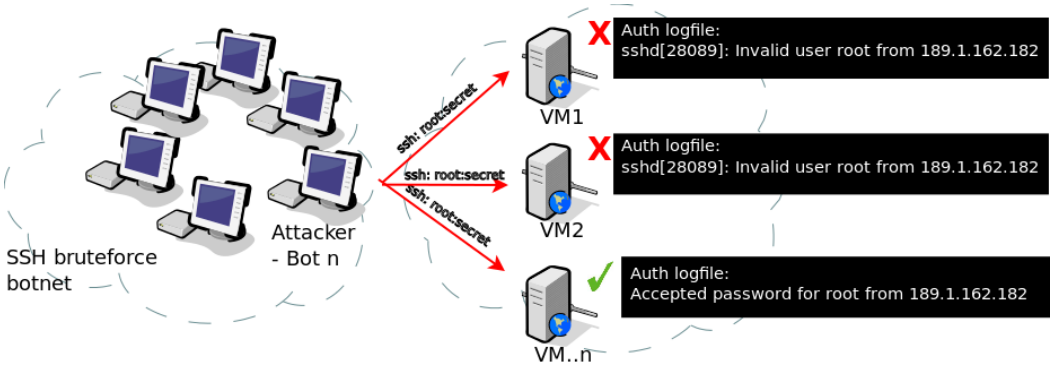


Figure 6.2: Distributed SSH bruteforce on weakly secured cloud instances

Several hypervisor vulnerabilities have been found in the past that allow an attacker to gain access to a cloud host from inside a VM (e.g. [108],[109]). As cloud computing and especially IaaS relies heavily on virtualization technologies this poses a threat to every cloud provider. By accessing the underlying host, an attacker not only gains access to real hardware, but also to all other VMs running on this machine. It is therefore essential, that VM escaping attacks get detected and appropriate counter measures are taken. This includes monitoring process activity on all cloud hosts for unusual activity, for example the commands that are being executed or syscalls as described by Hofmeyr et al. in [208]. The following risk scenarios for a rule based anomaly detection system have been identified:

- Misusage of VMs to execute follow-on attacks
- Hosting of (malicious) data or software
- Malicious creation / deletion of VMs
- Scalability attacks

6.3.3 Rule Based Anomaly Detection Architecture

Rule based anomaly detection can be already achieved by the presented SAaaS architecture (Chapter 4) and the Cloud Audit Policy Language (Chapter 5). Rules can be described as security policies with a corresponding threshold in CAPL format. Whenever a cloud sensor agent detect a change, which affects a user's instances, SAaaS agents get configured with the corresponding user's security policies and deployed to the affected VMs. The previously introduced "Attack on cloud scalability" scenario is used to explain an example of user-centric anomaly detection in more detail.
Consider a cloud user deploys an online shop within a cloud, see Figure 6.3. To share load, two web servers are running on different VMs, a load balancer is distributing incoming http requests. All web servers are combined to a virtual group "WWW-Cluster1" by a CAPL policy. Scalability features are enabled based on the metric "*http requests per second*". A CAPL upscaling policy ① regulates, that if the total number of incoming requests per second exceeds a certain threshold, e.g. 100 requests / second, an upscale event for group WWW-Cluster1 gets issued to the cloud management system and an additional web server VM gets created and added to WWW-Cluster1. To prevent attacks on scalability the following CAPL monitoring policy gets assigned to WWW-Cluster1:

Name : Scalability protection WWW-Cluster1

Policy Type : Scalability

VM Group : WWW-Cluster1

Metric : Requests per second

Threshold : 100

As a result, ② this policy gets registered at the SAaaS CMS sensor agent, which monitors CMS events. On an incoming upscale event for WWW-Cluster1, the defined

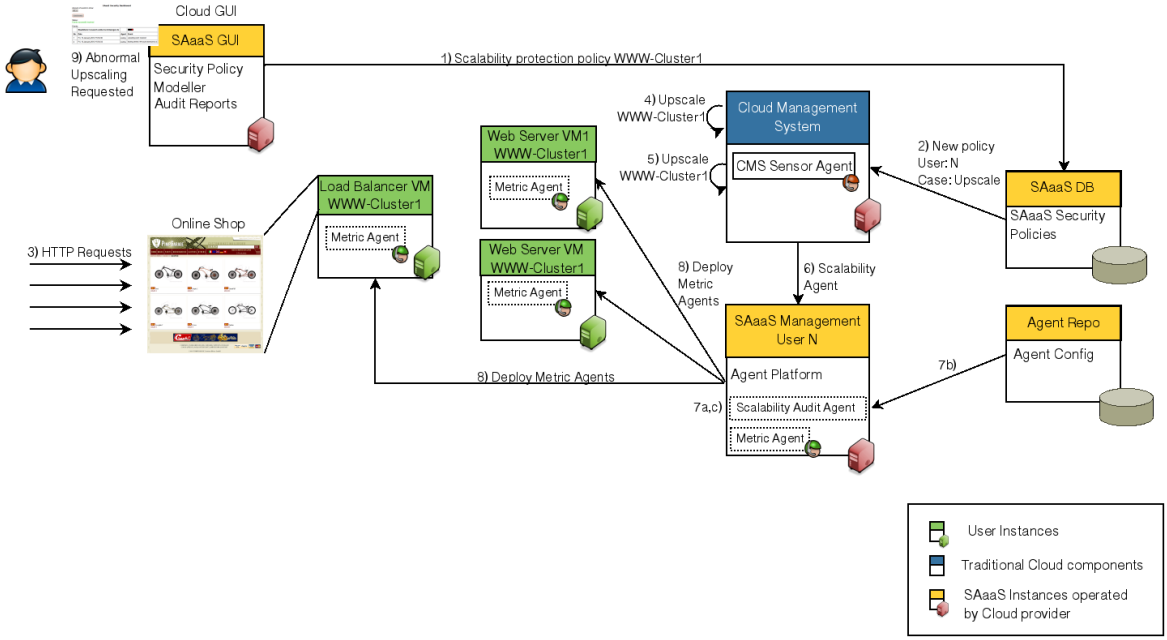


Figure 6.3: User-centric anomaly detection: cloud scalability

conditions are checked before execution of the upscale command.

Now let's assume there is a high load ③ on the web servers due to a product launch of the user's company. Therefore, the cloud management system gets an upscale event ④ for *WWW-Cluster1*, which gets intercepted by a SAaaS agent ⑤. The event provokes according to the policy set a concurrent audit, which leads to the creation and configuration of a new `Scalability Audit Agent` ⑥ at the user specific SAaaS Management VM. The audit agent gets created from the agent repository (7a), configured with the scalability check for WWW-Cluster1 (7b) and started (7c). The agent creates a new metric agent, which gets deployed to the load balancer to check the current total amount of registered requests ⑧. Additionally, metric agents get deployed to the web server VMs of WWW-Cluster1 ⑧ to validate the current load of HTTP requests. All metric agents report the result back to the audit agent. The audit agent evaluates the result and decides, dependent on the average load reported by the metric agents, if the upscale event is okay or not. The results get conditioned into an audit report. By pulling the information from multiple sources (load balancer

and WWW-Cluster1 web server VMs), a compromised load balancer or compromised agents at the web server VMs could be detected. In case the reported load is below the defined upscaling threshold this is considered as an anomaly, which get reported to the user ⑨. Different notification schemes, such as alerting the user via email, short text message or further actions are imaginable.

As for a cloud wide anomaly detection, the previously elaborated scenario "Cloud Login Bruteforce Attack" could be considered. In a similar manner as CAPL enables cloud user to define security policies for their VMs, it can be used for the cloud provider to define security policies for the whole cloud infrastructure. Thus, one could state: "If a successful login from one SRC IP address (the attacker) is preceded by a number of n failed login attempts, this is an abnormal behaviour, which needs to be reported. Figure 6.4 visualises the involved SAaaS components. It is assumed,

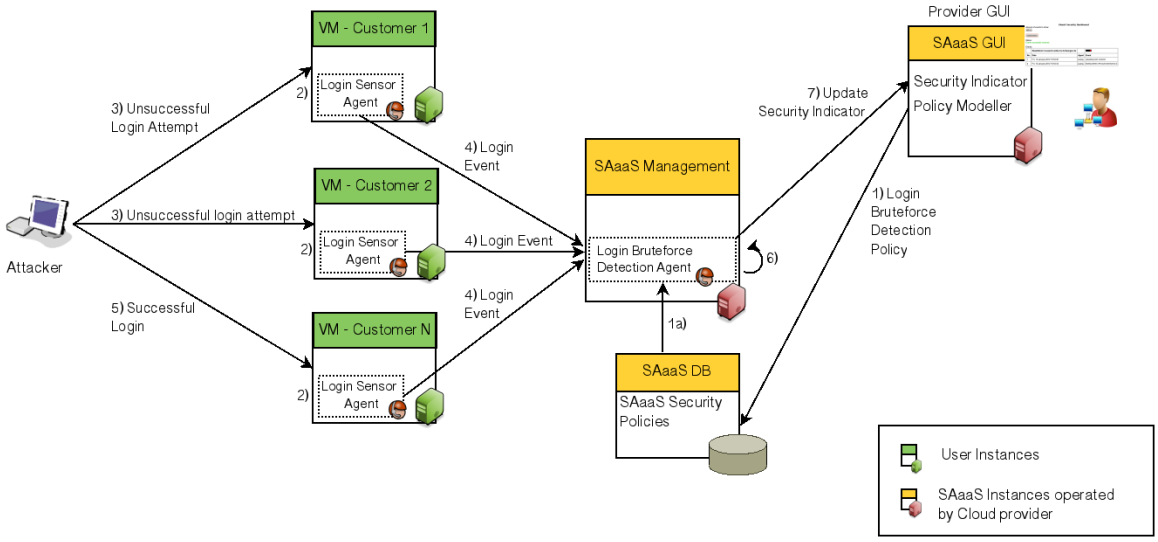


Figure 6.4: Cloud wide anomaly detection: Login bruteforce detection

that a corresponding CAPL policy was set by the cloud provider ①. Login sensor agents deployed over all VM instances ② report information ④ about (successful and unsuccessful) login attempts ③. In case a successful login ⑤ is detected, it is examined ⑥ against the login event history of this Src IP. In case the login violates the

defined security policy, an event is added to the SAaaS Security Event dashboard and the SAaaS security indicator is updated ⑦. Again, different notification or response schemes are imaginable.

These two scenarios were chosen to show, that whenever a rule can be defined to model an abnormal or normal behaviour, the combination of SAaaS monitoring agents and cloud audit security policies can be used to identify rule based detectable anomalies in a cloud infrastructure. However, as indicated in Table 6.1, not all cloud security risks can be addressed by a rule based detection method.

6.4 Behavioural Based Anomaly Detection

"Most current approaches to misusage detection involve the use of rule based expert systems to identify indications of known attacks. However, these techniques are less successful in identifying attacks, which vary from expected patterns" [209].
Similar risks, especially the ones originating from the increased complexity and multi-tenancy of a cloud infrastructure are not addressable by a static, rule based detection method. For those it is hard to define a meaningful security policy because of the complexity of the parameter space and the dynamic environment. Requirements for an anomaly detection system supporting these dynamics are:

- Cover a wide spectrum of different input parameters
- Learn how normal behaviour looks like
- Detect anomalies in this behaviour
- Adapt fast and robustly to the changing environment

To build up behaviour profiles, measurable key values called "input parameters" need to be identified, which can be linked with a cloud user's profile. Thus, a machine

understandable profile of his cloud usage behaviour can be built. Machine learning techniques can then be utilised to analyse these profiles. Important for the quality of the machine learning approach is the quality of the parameter describing a system, which are forming the input for the machine learning system.Thus, the identified cloud security issues (presented in Chapter 3.4) were analysed on their technical detectability and classified to either rule based or behavioural based detectability. This classification of cloud security issues is available electronically on the attached DVD in the folder "*CloudSecurityIssues*". All rule based detectable cloud security issues can be mitigated by modelling of CAPL security policies. For all remaining security issues, an analysis has been done, which information (input parameter) is necessary to support an anomaly detection in on this specific subject. Due to time constraints on finishing the PhD thesis, only input parameters for an anomaly detection in VM misusage were identified. As a result, Table 6.2 shows IaaS input parameters, which have have been identified by this research to be valuable for anomaly detection in IaaS clouds, and where this information originates. The presented SAaaS agents provide a well suited technical possibility to collect these information.

Classifier	**Description**	**Origin**
VM affiliation	The user id a VM belongs to	CMS agent
Time stamp	The time stamp of a VM management command	CMS agent
VM Action	The VM management command executed, e.g. VM-CREATE, VM-DELETE, VM-STOP, VM-MIGRATE	CMS agent
VM Creator SRC IP range	The IP range of a user accessing the CMS for VM management tasks, such as VM creation, VM altering, VM stopping	Agent monitoring cloud GUI logfile
Running VM Count	Number of currently running VMs of that user	CMS
VM services	Open ports indicating services offered on a VM	Audit agent

Continued on next page

Classifier	Description	Origin
VM image	The type of VM image chosen, such as Linux VM image, Windows VM image, OwnCloud or specific versions of those	CMS agent
VM parameter	Virtual hardware configuration of VM, such as Virtual CPU, Virtual RAM, Virtual disk space, amount of Virtual Networking interfaces, Virtual Network interface type, other Virtual peripherals	CMS
VM software	Software, which is installed within a VM. This could be derived from package management	Audit agent
VM Network protocols	Networking protocols used by a VM to communicate	Host agent
VM CPU consumption	CPU consumption utilised by VM	Host agent @ hypervisor
VM RAM consumption	RAM consumption utilised by VM	Host agent @ hypervisor
VM I/O consumption	I/O consumption utilised by VM	Host agent @ hypervisor
VM network bandwidth	Network bandwidth utilised by VM	Host agent @ virtual switch
Cloud data centre	Which target data centre is selected for VM, in case of multiple available cloud data centres, such as Amazon Availability Zones Zones[210]	CMS
VM outgoing network targets	IP range of destinations VM sends packets to	Host agent @ virtual switch
SRC IP range VM configuration	IP range of systems establishing configuration connections to VM, such as a direct SSH login or Windows Remote Desktop connection	Host agent @ virtual switch
Specific VM software metrics	Customised information of certain software installed on VM, such as web server load on web server VM	SAaaS metric agents

Table 6.2: Anomaly detection: direct input parameters for IaaS cloud usage

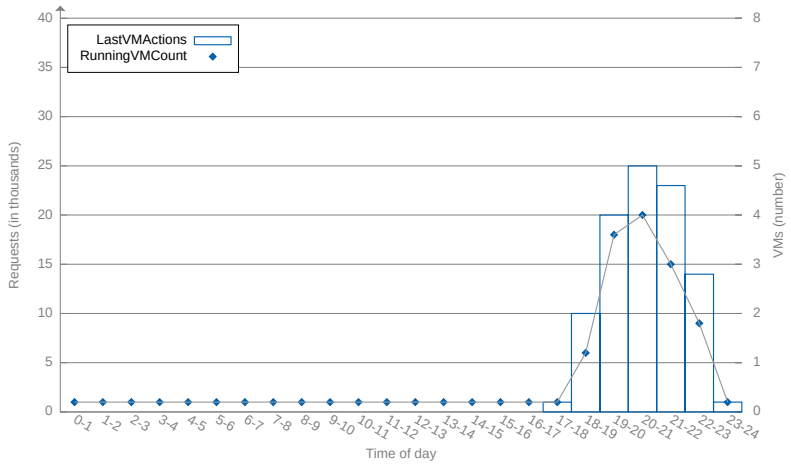


Figure 6.5: Anomaly Detection: online movie rental - normal behaviour

Based on those direct input parameters (information is directly available), additional derived parameters, called "features" [211, p2] can be deducted. Table 6.3 shows identified features for IaaS clouds. Features are based on the combination of multiple direct input parameters. They often indicate a trend of a certain action or behaviour, since they are referring to a certain time frame. The feature `Last VM count actions` contains an average number of VM management actions, such as VM-CREATE, VM-STOP or VM-MIGRATE over a time frame, for example the last hour. As an example, consider an online movie rental provider, who uses VMs in a cloud in addition to his own data center to absorb load peeks. Load peeks normally occur every day during the evening between 18:00pm and 24:00pm, since people are rather watching movies in the evening than during the day. As a result, the feature Last VM count actions is increasing until the peek of requested instances is reached and then decreases when people are finished watching, going to bed. Figure 6.5 shows a graphical representation of this normal behaviour. Additionally, to elaborate the difference between direct classifiers and features, the direct classifier `Running VM Count` is also shown. It follows the data line of the feature, since more VMs are running during the peek time and

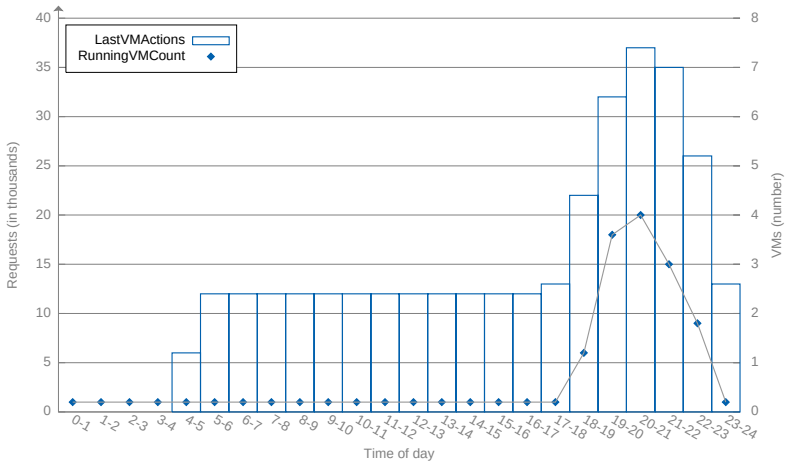


Figure 6.6: Anomaly Detection: online movie rental - abnormal behaviour

afterwards they get revoked[1]. So far, no additional value is achieved with the feature in addition to the direct classifier. Now, a malfunction in the cloud management system is assumed, causing a permanent creation and deletion of one virtual instance every five minutes, starting at 4:30pm. Figure 6.6 again, depicts the two classifiers. It can be seen, that the `Running VM Count` classifier does not expose the malfunction, since the average overall VM count does not change. However, the feature `Last VM count actions` exposes the malfunction, since the amount of VM actions (VM CREATE, VM-DELETE) within the time frame expands constantly. Naturally, an anomaly detection system is run not just for one day, but for multiple days, so this malfunction can get detected by comparison of the different days.

The target of the anomaly detection system for the Security Audit as a Service infrastructure aims to identify anomalies in IaaS clouds. This target scope is not limited to the detection of cloud abuse or attacks. The system is also valuable for cloud user and provider detecting anomalies not originating from a security incident, but a malfunction of a system component or misconfiguration. This could simply be a forgotten installation of a certain piece of software (e.g. anti virus software) or a configuration,

[1]It is assumed that one VM is always running, that's why there is always a VM count of one.

Classifier	**Description**	**Calculated From**
Last VM count create	Number of VM-CREATE commands of that user during a particular time frame	Time stamp & VM Action
Last VM count stop	Number of VM-STOP commands of that user during a particular time frame	Time stamp & VM Action
Last VM count delete	Number of VM-DELETE commands of that user during a particular time frame	Time stamp & VM Action
Last VM count migrate	Number of VM-MIGRATE commands of that user during a particular time frame	Time stamp & VM Action
Last VM count actions	Array of last VM management actions within a particular time frame	Time stamp & VM Action
Last VM count images	Array of last VM images chosen for a new VM within a particular time frame	Time stamp & VM Images
Last VM count VM parameters	Array of last virtual hardware chosen for a new VM within a particular time frame	Time stamp & VM parameter
...	...	...
Last *[DIRECT-CLASSIFIER-NAME]*	Array of *any* possible direct classifier listed in Table 6.2, within a particular time frame	Time stamp & *[DIRECT CLASSIFIER]*

Table 6.3: Anomaly detection: feature for IaaS Clouds usage

which is not coherent to a companies security policy. A main advantage of machine learning techniques, is their ability to automatically learn coherence's between different classifiers, which may, especially in a huge data set, not be evident for humans. For example, it can be easily identified by humans, that if suddenly multiple VMs of different cloud customers, are communicating over a so far unused network protocol at the very same time to a same target IP, this can be considered an anomaly, indicating a security incident, such as a botnet infection. Or these VMs are compromised to execute a DDoS attack, as already happened in the Amazon Cloud [79]. However, it is not that obvious to humans, if VMs are created from a compromised PC within a companies regular data center. The malicious VMs might be created from the usual SRC IP address, with the usual virtual hardware characteristics. However, the attacker might create the VM from a different VM image, e.g. an older, outdated version of the regular used image where security patches are missing. Or, the same image gets used but, to execute follow on attacks, an usually installed anti virus software is left

out. These fine grained combinations of different features can be learned by a machine learning technique, leading to an successful anomaly detection.

To address the introduced problem of behavioural based anomaly detection in an IaaS Cloud, a machine learning approach, based on artificial neural networks (ANN) was chosen. Neural networks are based on the concepts of statistical pattern recognition and have emerged as a practical technology, with successful application in many fields, such as speech recognition, fraud detection or classification of handwritten characters [211].

Same as for the Rule Based anomaly detection scenario, the behavioural based anomaly detection is split into "User-centric anomaly detection" and "Cloud wide anomaly detection" sub-scenarios. The following scenarios are developed.

6.4.1 User-centric Anomaly Detection

User anomaly scenarios are focused on the behaviour analysis of one particular cloud user. The target is to learn the cloud usage behaviour of this particular user and to detect anomalies in case his behaviour changes. By monitoring user account activity, user cloud interaction, such as interaction time or origin of requests, VM images used, VM instance behaviour, such as CPU utilisation, memory consumption, network I/O, time of VM utilisation, etc., a user's cloud usage behaviour can be learned. Thus, abnormal usage scenarios can be identified. The following use cases for user specific behavioural based anomaly detection are addressed:

The possibility to quickly aggregate massive loads of computing power is very attractive. This is also true for criminals who can use this power for example to crack hashed passwords or decode CAPTCHAs. Technically there is however no non-intrusive way to decide for what exactly cloud computing power is used. Even if there was, data protection laws and credibility concerns prevent such monitoring. However, monitoring the user's behavior for anomalies can help detecting account misusage. For

example heavy activity on a rather silent account would cause suspicion, especially if the login source is from another country or continent. Additionally, to the VM count and geographical login source, the weekday and time as well as usage of available VM types can be taken into account.

Anomaly Detection Model For User Scenarios

While rule based anomaly detection is a rather simple task, whenever a meaningful threshold can be applied, behavioural based anomaly detection can be used whenever a threshold definition is not possible. This is a rather challenging topic. Depending on the specific use case scenario, different input information, *classifiers*, need to be used to learn a certain usage behaviour and successfully detect anomalies. To show the feasibility of a behaviour based anomaly detection system, this research concentrates on one specific risk: Account misusage. It aims to answer the following research question: *"How can it been verified, that a user is responsible for a certain VM creation?* The anomaly detection system needs to learn, how a user uses a particular VM by analysing create and stop times of it over a certain training period. Afterwards, the system is able to tell, based on a single VM-CREATE or VM-STOP event, if this fits to the user's behaviour or not.

To show the applicability and performance of an anomaly detection system, an IaaS cloud usage data set - the training data - was needed to test the system's detection rate. Quality of training data is important, since if the training data is more accurate, the performance of the trained system will be improved. Collecting training data can be achieved in three ways: recording real cloud usage data, using sanitised data or using simulated data [212]. Multiple public cloud providers, such as Amazon or IBM were contacted, but refused to disclose usage data, not even in an anonymized form. Thus, a simulation environment for IaaS cloud usage was created (presented

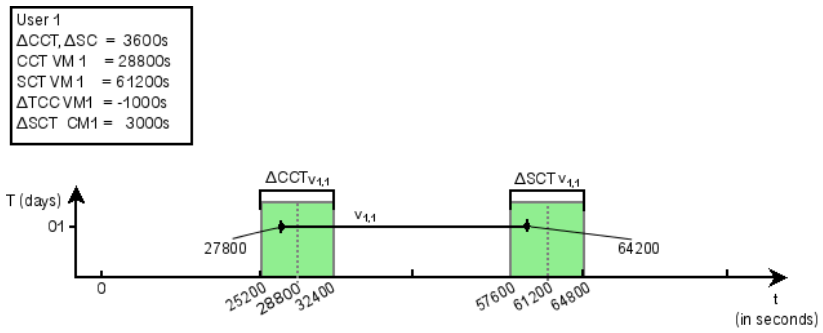


Figure 6.7: Calculation of a VM's runtime

later in this chapter). The following merely models are utilised to show the detection performance. They are created from experience with the university-wide cloud infrastructure CloudIA at Furtwangen University and multiple discussions with a German cloud provider. For the use case of user-centric behavioural anomaly detection the following normal behaviour is assumed.

Normal Behaviour

A cloud user is using one VM each day to work with. He starts up the VM at a certain time, e.g., 8:00 AM and stops it at a certain time, e.g., 5:00 PM. The start of the VM results at the cloud management system in a *VM-CREATE* event, a stop of a VM is registered as a *VM-DELETE* event. In a perfect scenario, a user would create and stop his VM always at the very same time every day. But this is not true for real world scenarios. Thus, the real VM create time and VM stop time gets calculated by a VM core create time (CCT) + a variation time value ΔCCT defined by random and a VM core stop time (CST) + a random value ΔCST. The specification of ΔCCT and ΔCST is important, since it is a control parameter for the accuracy of the detection of the neural network. If it is defined either very loosely, or too narrow, detection rate of the anomaly detection system is affected, since differentiation between normal and abnormal behaviour becomes less identifiable. Thus, this parameter has been implemented as a configurable variable, see later Section 6.4.3. For this work, a

default of 60 minutes is always assumed, since it provides good results. However, for a quantitative performance evaluation of the anomaly detection system, this parameter and its impact on the overall detection rate needs to be considered. Due to time constraints on finishing the PhD thesis, this has been left out for future work.
Figure 6.7 visualises these relations for one user for one VM by using the following example data, which are reused in all scenarios:

- VM1-CREATE: CCT+ $\Delta CCT = 28.800s - 1.000s = 27.800$s (7:43am)
- VM1-STOP: CST+ $\Delta CST = 61.200s + 3.000s = 64.200$s (5:50pm)

All values are measured in seconds. Start and stop times are seconds of a day, starting from 00:00 AM. This results in final VM create time of 27800s and a final VM stop time of 64200. The normal behaviour of a user can be described by the following mathematical model: A user U has one VM. The VM's runtime is defined by a Core Create Time (CCT) and a Core Stop Time (CST):

$$U\,VM = (CCT, CST)$$

CCT and CST are measured in seconds of a day between 00:00:00 and 23:59:59:

$$0 \leq CCT, CST \leq 86400 \; with \; CCT, CST \in \mathbb{N}$$

where

$$CCT < CST$$

The user's VM behaviour is monitored over multiple training days D:

$$D_t = \{D_1, ..., D_n\} \; with \; t \in \mathbb{N}$$

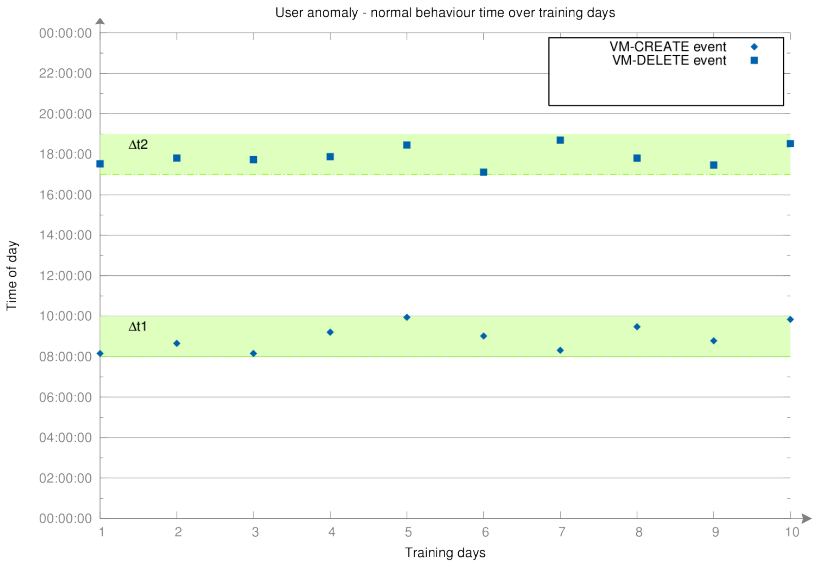


Figure 6.8: User Anomaly - Normal Behaviour time over training days

As introduced, real VM create time and VM stop time gets calculated by the VM core create time (CCT) + a random time value ΔCCT and a VM core stop time (CST) + a random value ΔCST where its range is defined by a positive or negative R:

$$-R \leq \Delta CCT, \Delta CST \leq +R \; with \; R \in \mathbb{N}$$

R unequal CST - CCT

Thus, the user's VM runtime is calculated by:

$$UVMD_t = (CST + \Delta CST_t) - (CCT + \Delta CCT_t)$$

In this work, a linear distribution of CMS events for each user is considered. Table 6.4 shows an example data set of the normal behaviour of one user, creating and stopping one VM over ten training days, with a normal deviation time ΔCCT and ΔCST of 120 minutes. The row "Events" indicate a VM-CREATE or VM-STOP event, row "Time" shows the corresponding time stamp in a human friendly format.

	Day 1		Day 2		Day 3		Day 4		Day 5	
VM Event	C	D	C	D	C	D	C	D	C	D
Time	08:17:33	18:39:01	07:56:49	18:48:36	08:49:48	17:10:20	08:05:54	17:35:11	08:58:51	17:23:33

	Day 6		Day 7		Day 8		Day 9		Day 10	
VM Event	C	D	C	D	C	D	C	D	C	D
Time	08:01:22	17:17:03	09:20:57	18:25:56	08:45:23	18:15:56	08:53:48	17:02:47	07:44:41	16:54:07

Legend:

C – VM-CREATE event

D – VM-STOP event

Table 6.4: Example data set, normal behaviour: one user, one VM over ten training days

Based on these raw input data, Figure 6.8 shows a detailed graphical representation of VM-CREATE and VM-STOP events for the introduced normal behaviour for each day over the ten training days. However, for anomaly detection, a more useful representation is depicted in Figure 6.9, which visualises the event count over ten training days. Again, normal deviation times ΔCCT and ΔCST are indicated by the green rectangles.

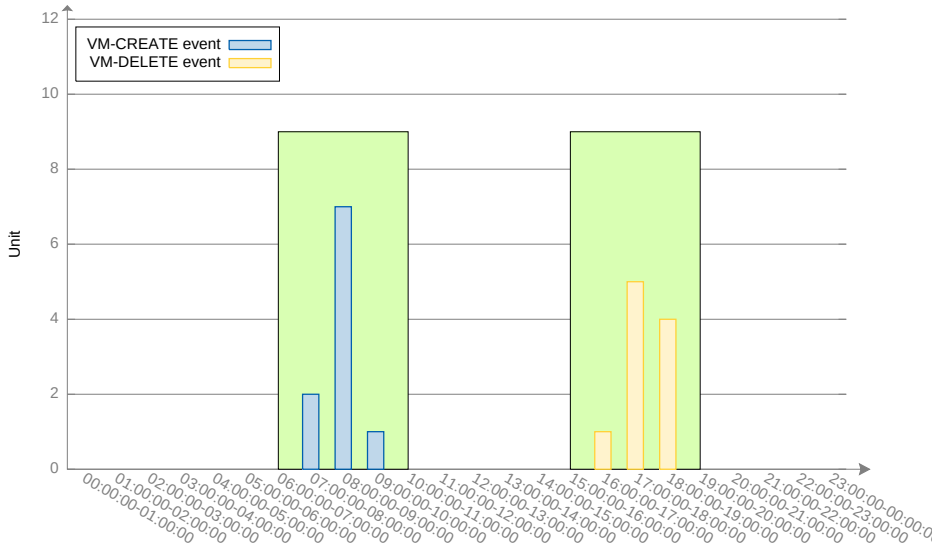


Figure 6.9: User-centric anomaly detection - Normal user behaviour

6.4.2 Cloud Wide Anomaly Detection

Whereas the prior scenario is focused upon the behaviour analysis of one single cloud user, the cloud provider's point of view is also very interesting. Its main target is to detect cloud usage changes within a huge amount of completely differently acting cloud users' instances, which could indicate a security incident. This could happen in case an attacker scans over a cloud IP range, infects loosely secured cloud instances and misuses them for follow-on attacks, such as a Distributed Denial of Service attack. This use case is already introduced in detail in Section 6.3.2. It also applies to the cloud wide behavioural anomaly detection, since a change in communication behaviour of the compromised VMs examined from a cloud wide point of view can detect such a security incident.

Furthermore, the possibility to quickly aggregate massive loads of computing power is very attractive as introduced earlier. This also results in a changed VM usage behaviour, probably over multiple instances of the same or different cloud customers, which can be detected by a cloud wide behaviour based anomaly detection system.

The following risk scenarios for a behaviour based anomaly detection system have been identified:

- Account misusage
- Hosting of (malicious) data or software
- Malicious creation / deletion or misusage of VMs

This list is indicative and not definite. The risks have been chosen from the identified cloud security issues, presented in *Chapter 3.4 - Cloud Computing Security Issues* to demonstrate the feasibility of the system.

Cloud Wide Behaviour Model

The cloud wide behaviour detection is build upon the user specific model, but now multiple users are considered. Furthermore, it is assumed, that each user uses a different amount of virtual machines. Each single virtual machine has its own specific run time. This adds two more dimensions of complexity (userid, normal VM count) to the scenario, compared to the presented user-centric anomaly scenario.

Normal Behaviour

Normal behaviour is described as: a cloud provider runs an Infrastructure as a Service cloud. The provider has multiple customers, with a minimum of 1. Every customer uses a different amount of VMs with individual start and stop times of each VM. The number of VMs a user is normally using each day does not underlie big fluctuations. The scenario can be described by the following mathematical model: A provider has multiple cloud users u with a minimum of 1, described by the index set:

$$U_u = \{U_1, ..., U_n\} \ with \ u \in \mathbb{N}$$

Each user has multiple VMs, indexed by whole numbers i:

$$VM_i = \{\, VM_1, ..., \, VM_n\} \ with \ i \in \mathbb{N}$$

Each single VM's runtime is defined by a CCT and a CST:

$$U_u VM_i = (CCT_{ui}, CST_{ui})$$

CCT_{ui} and CST_{ui} are measured in seconds of a day between 00:00:00 and 23:59:59:

$$0 \leq CCT_{ui}, CST_{ui} \leq 86400 \ with \ CCT_{ui}, CST_{ui} \in \mathbb{N}$$

where

$$CCT_{ui} < CST_{ui}$$

The users' VMs behaviour is monitored over multiple training days D:

$$D_t = \{D_1, ..., D_n\} \ with \ t \in \mathbb{N}$$

Again, the real VM create time and VM stop time gets calculated by the VM core create time (CCT) + a random time value ΔCCT and a VM core stop time (CST) + a random value ΔCST where its range is defined by a positive or negative R:

$$-R \leq \Delta CCT, \Delta CST \leq +R \ with \ R \in \mathbb{N}$$

Thus, the final create and stop time of a virtual machine VM_i of a user U_u at a specific training day D_t is defined by:

$$U_u VM_i D_t = (CST_{ui} + \Delta CST_{uit}) - (CCT_{ui} + \Delta CCT_{uit})$$

An an example the following simulation parameters are assumed: 10 training days, 1000 user, User 3 uses 5 VMs every day, CCT of VM 2: 28800s, CST of VM 2: 61200, at training day 7: ΔCCT: -1000, ΔCST: 3000. This result in the following:

$$D_t = 1, ..., 10$$

$$U_u = 1, ..., 1.000$$

$$VM_i = \{VM_{3_1}, VM_{3_2}, VM_{3_3}, VM_{3_4}, VM_{3_5}\}$$

$$VM_2 - CREATE = CCT_{3,2} + \Delta CCT_{3,2} = 28.800s - 1.000s = 27.800s \ \textit{(7:43am)}$$

$$VM_2 - STOP = CST_{3,2} + \Delta CST_{3,2} = 61.200s + 3.000s = 64.200s \ \textit{(5:50pm)}$$

$$U_3 VM_2 D_7 = (27.800, 64.200)$$

Thus, user 3 with 5 VMs is represented as:

$$U_3 = \{ VM_{3_1}, VM_{3_2}, VM_{3_3}, VM_{3_4}, VM_{3_5} \}$$

Figure 6.10 tries to depict the resulting data set for ten users. It can be seen, that a meaningful graphical representation of the CloudWideAnomaly scenario is not possible anymore, as it consists of four variable dimensions. Details on the data set will be presented in Section 6.4.3. The detailed data values are available on the attached DVD in folder *CloudUsageSimulator/DataSets/provider anomaly01-simutestuser_Sorted_1000User.csv*.

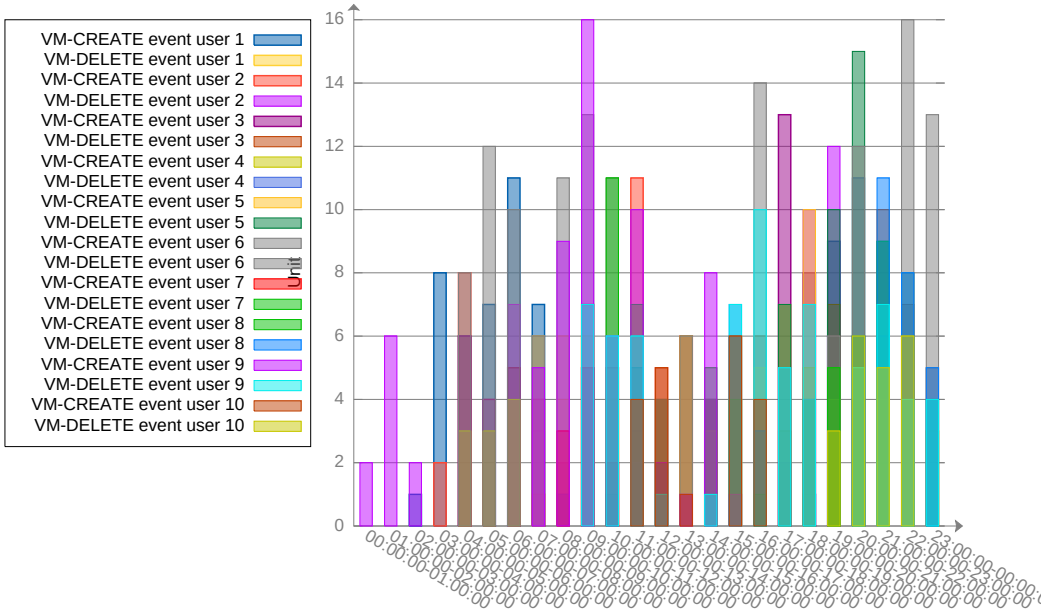


Figure 6.10: CloudWideAnomaly01 scenario data set

6.4.3 Anomaly Detection Architecture

The anomaly detection system presented in this work is a sub component of the Security Audit as a Service architecture for IaaS cloud environments. The source for the anomaly detection system presented in this work are SAaaS agents, monitoring events, which build classifiers, as introduced in *Table 6.2 - Direct input parameters* and *Table 6.3 - Features* earlier in this Chapter. The network is trained with historical user data and evaluates the newly received event if it indicates an anomaly or not. For the anomaly detection in IaaS clouds a supervised learning algorithm with a feed forwarded net has been used. For creation of the neural network, training and testing the neural network editor Membrain [213] was used. Data sets of cloud usage are simulated, thus the simulation environment gets described first.

Cloud Usage Simulator

To test the presented anomaly detection scenarios, an IaaS cloud usage data set was required. As mentioned earlier, real world data from IaaS cloud providers were not accessible. Thus, a simulation environment was created to simulate the presented normal cloud usage behaviour and anomaly scenarios.

Cloud usage data set

For each simultation run, first a normal behaviour data set is generated. Since the main target of this research stage is to evaluate, that anomaly detection based on neural networks provides an extra value to improve security in IaaS clouds, three user-centric and one cloud wide anomaly scenario is developed and implemented. However for simplicity, not all introduced direct parameters and features introduced in Table 6.2 and Table 6.3 were implemented. Only the following parameters were implemented.

- Direct input parameters
  - VM affiliation (User id)
  - VM Action
  - Time stamp
  - Running VM Count
- Features
  - Last VM Count Create
  - Last VM Count Delete

With those input parameters, the following indicative list of anomaly scenarios are developed. They are a selection of more possible scenarios and were chosen to demonstrate the feasibility of the anomaly detection approach.

- **User-centric anomaly scenarios**
  - UserAnomaly01 - Abnormal VMs outside of normal VM usage time
  - UserAnomaly02 - Abnormal VMs within normal VM usage time
  - UserAnomaly03 - Abnormal VMs within normal VM creation time frame (ΔCCT) and stopped within normal VM stop time frame (ΔCST)
- **Cloud wide anomaly scenarios**
  - CloudWideAnomaly01 - Within a huge number of differently behaving cloud users, the behaviour of one user changes drastically.

Within those scenarios, an abnormal VM is considered a VM, which is not intentionally created by the original cloud customer.

UserAnomaly01

In scenario UserAnomaly01, an increasing number of VMs gets created outside the working hours. Figure 6.11 shows a graphical representation of the corresponding data file. Abnormal VMs are depicted in red, normal behaviour is distributed within the green area. In this example data set a total of ten abnormal VMs were created.

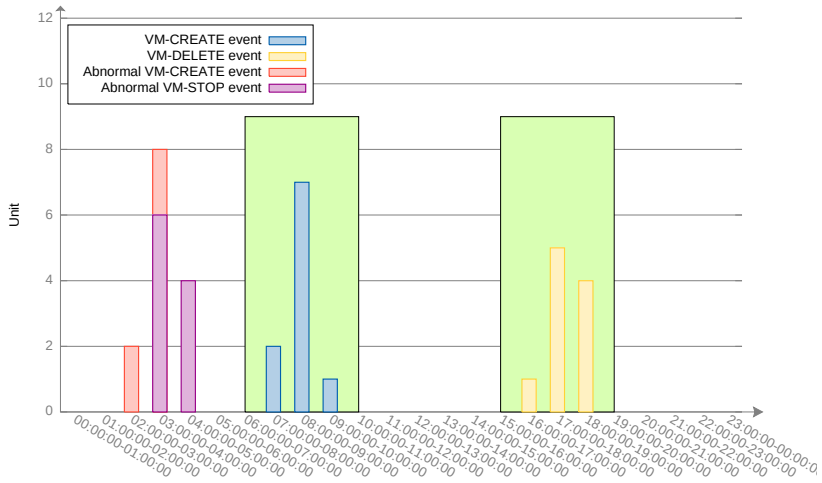


Figure 6.11: UserAnomaly01 scenario data set

Two VMs were created between 02:00:00 - 03:00:00 and eight VMs between 02:00:00 - 03:00:00. The VMs were shutdown sequentially, six between 02:00:00 - 03:00:00 and four between 03:00:00 - 04:00:00.

UserAnomaly02

As a next step, in scenario UserAnomaly02 the abnormal VM-CREATE and VM-STOP events are created within the "normal" VM usage hours as depicted in Figure 6.12. Again, ten abnormal VMs were created, this time three between 11:00:00 - 12:00:00, four between 12:00:00 - 13:00:00 and four between 13:00:00 - 14:00:00. Two were stopped between 12:00:00 - 13:00:00, two between 13:00:00 - 14:00:00 and six between 13:00:00 - 14:00:00.

UserAnomaly03

In the final user-centric anomaly scenario, UserAnomaly03, the abnormal VM-CREATE and VM-STOP events are generated during the normal time period the user would start and stop his VMs, thus within ΔCCT and ΔCST. A resulting example data set is pictured in Figure 6.13. This time the bars of the normal and abnormal event

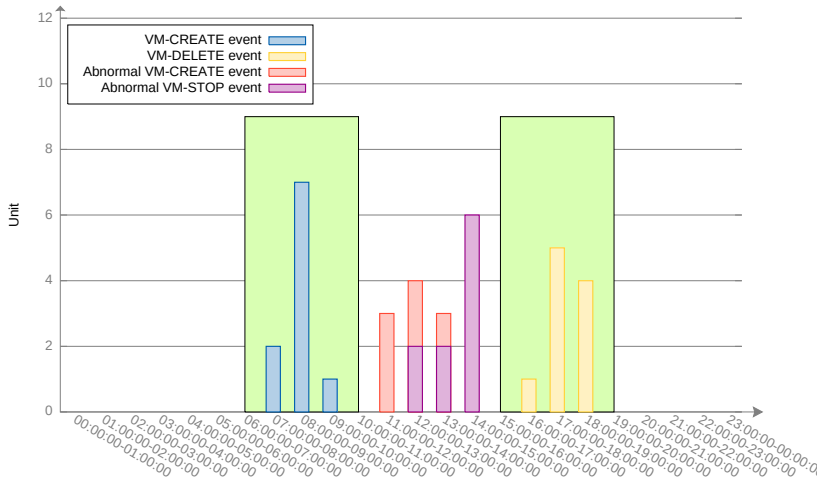


Figure 6.12: UserAnomaly02 scenario data set

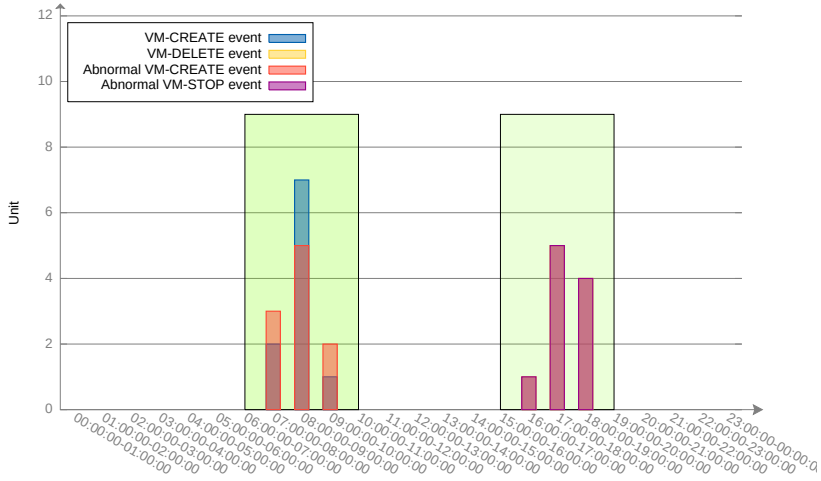


Figure 6.13: UserAnomaly03 scenario data set

are lying congruent on each other. By just looking at a VM Action time behaviour, it would be not possible to tell, which VMs are the normally created ones, and, which one are the abnormally created ones. In this scenario three abnormal VMs were created between 07:00:00 - 08:00:00, five between 08:00:00 - 09:00:00 and two between 09:00:00 - 10:00:00. The abnormal VMs were stopped with exactly the same behaviour as the normal ones: one between 16:00:00 - 17:00:00, five between 17:00:00 - 18:00:00 and four between 18:00:00 - 19:00:00.

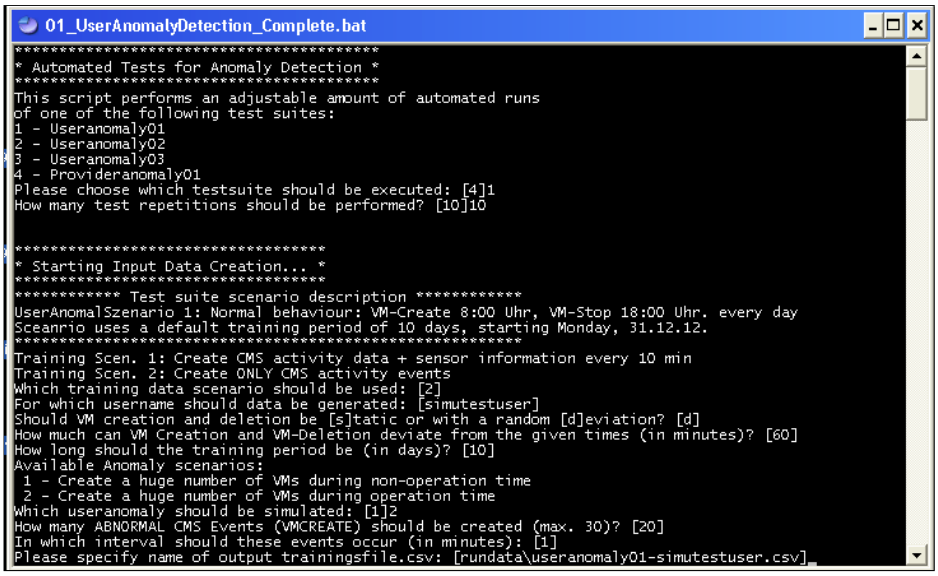


Figure 6.14: Cloud usage data simulator

CloudWideAnomaly01

This scenario simulates a cloud wide anomaly scenario. For normal behaviour data, multiple users are simulated. Each user differs in his cloud usage behaviour. Each user uses a variable amount of VMs, with different VM start and VM stop times. Abnormal data are simulating a change in the behaviour of one, randomly chosen user. As elaborated earlier in Section cloud Wide Anomaly Detection - Normal Behaviour, a meaningful graphical representation of this use case is not feasible any more, since a four dimensional variable space of input classifier exists: UserID, VM count of normally used VMs each day, VM create time, VM stop time.

Simulation Process

The developed cloud simulator is based on multiple python scripts working together with the neural network tool Membrain. Figure 6.14 shows a screen shot of the simulator's entry mask. To support the presented scenarios, the following simulation parameters are collected before simulation by a textual wizard:

1. Simulation scenario (UserAnomaly01, 02, 03, CloudWideAnomaly)

2. Number of simulation runs (R)

3. Normal data attributes:

   a) Random deviation range of daily VM-CREATE and VM-STOP commands (ΔCCT, ΔCST)

   b) Number of training days

   c) Max. number of VMs a user can normally have (CloudWideAnomaly)

4. Anomaly data attributes:

   a) Number of abnormal CMS events (VM-CREATE or VM-STOP) to be generated

   b) Number of additional VMs, which are still considered normal

Listing 6.1 shows a cut out of the data set of user U_5 for normal behaviour in scenario B - CloudWideAnomaly01. It shows the userid (U), the time stamp (TStmp) of the monitored event, the type of the event (VME), where C = VM-CREATE, D = VM-DELETE, the running VMCount ($VMCount$), the history of issued VM-CREATE ($LastVMCC$) and VM-DELETE ($LastVMCS$) events on that training day and if this event is considered to be an anomaly ($Anomaly$) or not. Before the data set is fed to the the neural net, the data get normalised to a range between 0 and 1 (as is normal practice for neural networks [211, p21]).

```
1 U;   TStmp; VME; VMCount; LastVMCC; LastVMCS; Anomaly
2 5;   15289; C;   1;       1;         0;         0
3 5;   24355; D;   0;       1;         1;         0
4 5;   14335; C;   1;       1;         0;         0
5 5;   30507; D;   0;       1;         1;         0
6 5;   14060; C;   1;       1;         0;         0
7 5;   26095; D;   0;       1;         1;         0
8 5;   14237; C;   1;       1;         0;         0
9 ...
```

Listing 6.1: Data set normal behaviour scenario CloudWideAnomaly01

After input selection is done, the following steps are automatically performed during a simulation run:

1. Generation of a data set containing normal and abnormal behaviour events according to the scenario chosen
2. Splitting of data set (randomly) into training data (80%) and a testing data (20%)
3. Loading and training of neural network in Membrain
4. Running the testing data set through the trained network
5. Evaluation of the testing data set
6. Storing of results to test run specific result file

The steps listed above, are performed automatically and form one simulation run R. To validate, that the results of one simulation run are reproducible and the network's detection performance is steady, a number of multiple simulation repetitions n, e.g. ten can be defined. Multiple simulations are also required to account for any bias in the simulation data set. Thus, n runs of the simulation sequence listed above are sequentially performed. Each repetition goes through the whole simulation sequence, including new training and testing data generation and randomisation. The overall results of each single simulation run ($R_1, R_2, ..., Rn$) are combined into one comma separated value file (e.g. useranomaly03-result-evaluated-2013-10-04-20-35-53.csv). After all simulation repetitions are done, the overall result file (e.g. (e.g. useranomaly03-result-evaluated-overall-2013-04-15-16-49-43.csv) is automatically opened in a spreadsheet program, depicted in Figure 6.15.

The sequence diagram in Appendix A.6 shows the different python and Membrain scripts involved in the simulation of scenario UserAnomaly01. The data set for normal and abnormal behaviour gets created by the entry python script

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		Run	Training Repititions	Net Error after Training	Total Testing Patterns	Correct Detected	False Detected	Error Rate of Detection in %	Single Run Data File					
2		1	948	0	8	8	0	0	results\useranomaly03-result-evaluated-2013-10-04-20-35-53.csv					
3		2	402	0	8	7	1	12,5	results\useranomaly03-result-evaluated-2013-10-04-20-35-47.csv					
4		3	469	0	8	7	1	12,5	results\useranomaly03-result-evaluated-2013-10-04-20-35-45.csv					
5		4	890	0	8	8	0	0	results\useranomaly03-result-evaluated-2013-10-04-20-35-42.csv					
6		5	951	0	8	8	0	0	results\useranomaly03-result-evaluated-2013-10-04-20-35-37.csv					
7		6	489	0	8	7	1	12,5	results\useranomaly03-result-evaluated-2013-10-04-20-35-32.csv					
8		7	152	0	8	8	0	0	results\useranomaly03-result-evaluated-2013-10-04-20-35-30.csv					
9		8	175	0	8	6	2	25	results\useranomaly03-result-evaluated-2013-10-04-20-35-28.csv					
10		9	210	0	8	8	0	0	results\useranomaly03-result-evaluated-2013-10-04-20-35-27.csv					
11		10	499	0	8	7	1	12,5	results\useranomaly03-result-evaluated-2013-10-04-20-35-26.csv					
12														
13	**Average***	**8**	**510,25**	**0**	**8**	**7,25**	**0,75**	**9,375**						
14														
15	*(without min & max)													
16														

Figure 6.15: Anomaly detection evaluation

`UserAnomaly01_Complete` *Workflow_Automated*. The main simulation is implemented as multiple python scripts. Data splitting is done randomly by a Membrain script, which then continues with training the neural network until the target net error is reached. A video of an automated simulation of scenario UserAnomaly03 with ten simulation runs is available at the attached DVD to this thesis in folder or online available at: https://doeli.de/web/video/ SimulationUserAnomalyScenario3 Frank-Doelitzscher.mov.

6.4.4 Neural Networks for Anomaly Detection

The data sets created by the cloud simulator are fed to a neural network, which trains and learns the simulated behaviour. In the following, the neural networks developed for the several presented anomaly detection scenarios are presented.

UserAnomaly01

The scenario UserAnomaly01 should detect if a VM was used at an abnormal time a user normally uses his VM. Figure 6.16 shows a graphical representation of the neural network used for scenario UserAnomaly01.

It consists of one input layer 2 hidden layers and 1 output layer. The hidden layers contain ten neurons each. This number proved to be working best during multiple

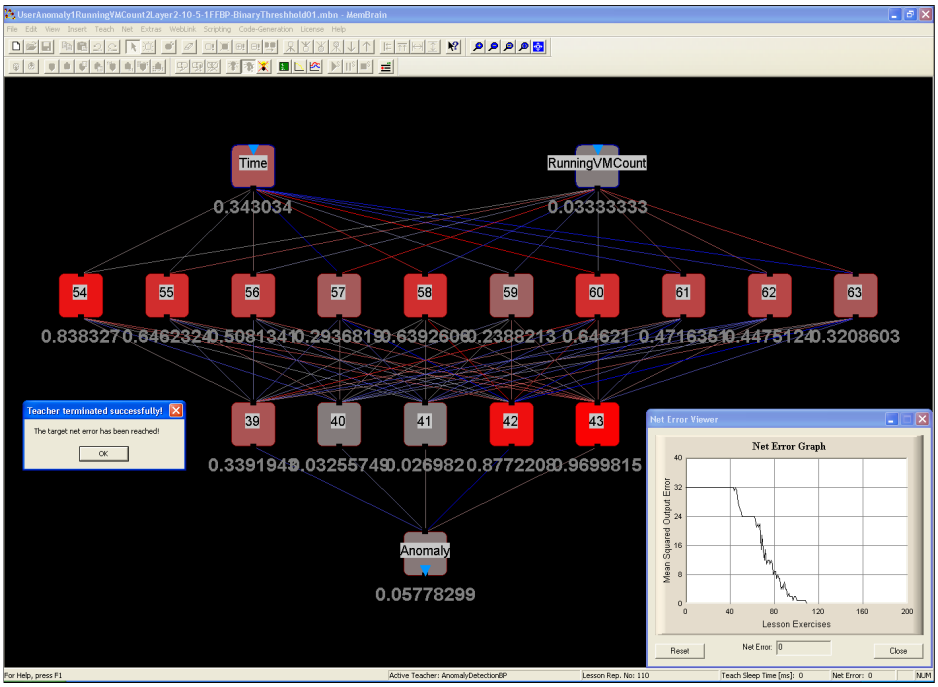


Figure 6.16: Feed forward net for scenario "UserAnomaly01"

simulation runs in terms of detection rate and learning speed. The following input classifier were chosen:

- Time stamp
- Running VM count

Time stamp represents the time a CMS event is received, Running VM Count corresponds to the amount of currently running VMs on a specific day. One output neuron indicates if the processed input information (VM-CREATE or VM-STOP event) is considered abnormal or not. For the learning algorithm a Backpropagation network is used since it is typically used for pattern recognition [214]. "Binary by threshold" error calculation with a threshold of 0.1 and no further calculation was used. A target net error of 0.001 was defined, meaning that the net will stop training when the target net error has been reached. Before training the net is always reset and weights are randomised. Figure 6.16 shows a graphical representation of the net after a successful training period with its weights and the net error graph. Again, the learning curve of

that simulation run is displayed as the net error graph and the target net error has been reached after 110 learning repetitions (shown at bottom).

UserAnomaly02

Scenario UserAnomaly02 covers the case, where abnormal VM(s) are created during the time a user normally uses his VM. Since this scenario will use anomaly data, where an attacker misuses the user's VM within his normal VM usage time (see Figure 6.12), two additional input classifiers LastVMCountCreate and LastVMCountDelete are added, building the following input feature sets:

- Time stamp
- Running VM count
- LastVMCountCreate
- LastVMCountDelete

LastVMCountCreate describes the amount of VM-CREATE actions already executed on this day. LastVMCountDelete describes the VM-DELETE actions executed at this day. All other network characteristics are the same as described in Section 6.4.4 above.

UserAnomaly03

Scenario UserAnomaly03 finally covers the case where abnormal VMs are created within the specific time frame a user normally creates and stops his VM Thus, the run time of abnormal VMs is similar to the normal ones. The same network as for UserAnomaly02 scenario was used. Figure 6.17 shows a graphical representation of the net after a successful training period with its weights and the net error graph.
The approximation and balancing of the different neurons during training phase is depicted in Figure 6.18. It shows the deviation between the net's trained (blue curve) and expected (red curve) result of the output neuron between training run 14 and 21. It can be seen, that during those runs, the neurons' weights are getting adjusted, so that the calculated results gets closer to the expected result. An over-evaluation

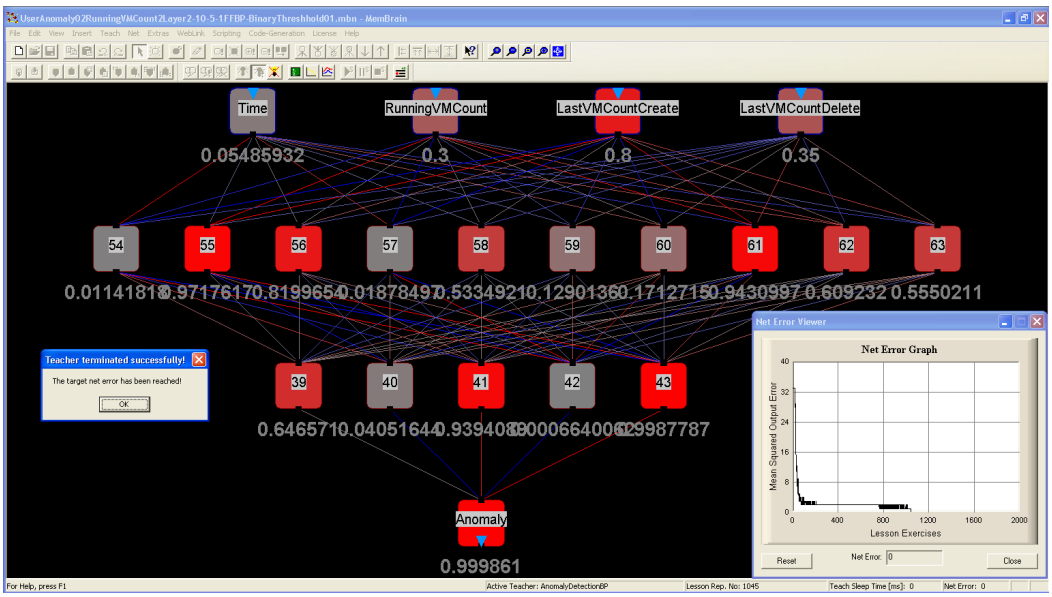


Figure 6.17: Feed forward net for scenario "UserAnomaly02" and "UserAnomaly03"

(around pattern 17) gets readjusted so that the target net error is reached (around pattern 19).

Cloud Wide Anomaly Detection with Neural Networks

For the detection of cloud wide anomalies the target is to identify a changing user behavior within a huge set of differently acting users. The neural net from scenario UserAnomaly03 was extended by the input classifier userid, forming the following input classifier set:

- Userid
- Time stamp
- Running VM count
- LastVMCountCreate
- LastVMCountDelete

Figure 6.19 depicts the neural network. All other network characteristics are the same as in scenario UserAnomaly03. The source code of the simulator is available on the thesis DVD in folder "CloudUsageSimulator". Next, simulation results of the presented anomaly detection system are presented.

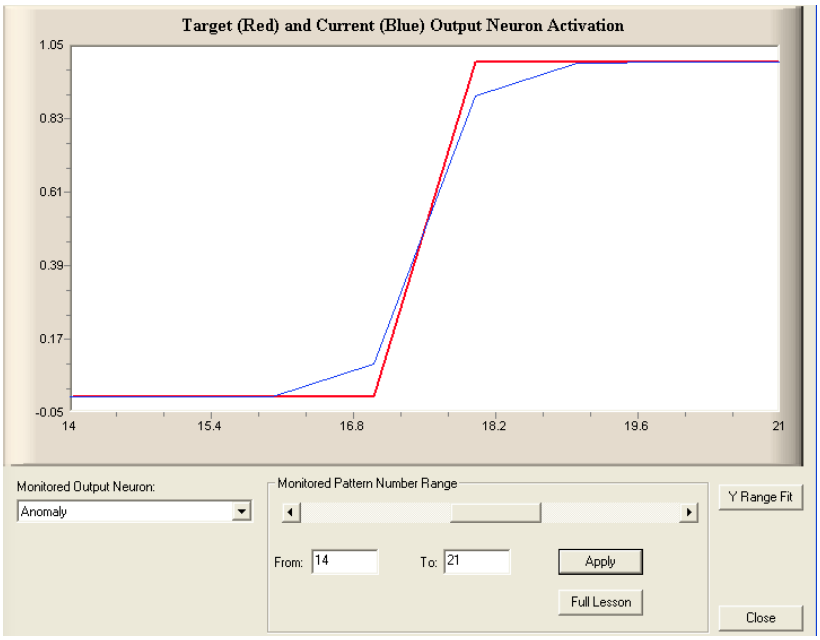


Figure 6.18: Neural networks learning curve in scenario UserAnomaly03

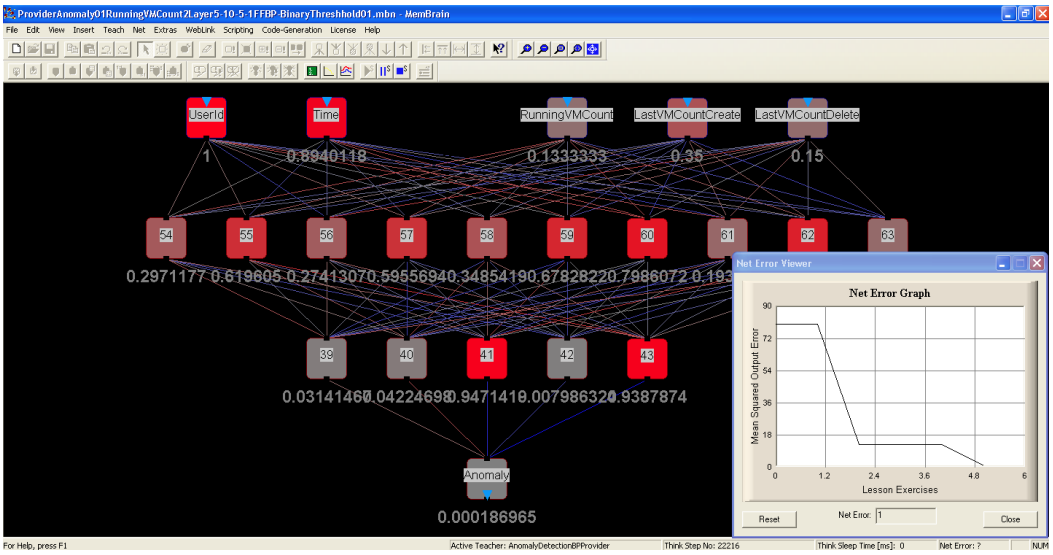


Figure 6.19: Feed forward net for scenario “CloudWideAnomaly”

6.5 Results

All simulations were performed on the same hardware, listed in Table 6.5. Each presented scenario was simulated at least 100 times to evaluate the detection results. A target net error of 1 was aspired during the learning phase of the network. As a target result, an average error below 10% was aimed for all scenarios, since this appears to be a widely accepted error rate in pattern recognition [211]. Fine tuning of the network took place, to optimize the target results.

System	Virtualbox VM
CPU	2x Intel Core i7-2675QM @ 2.20GHz
RAM	1562 MB
OS	Windows XP
Membrain Version	5.00 01 00
Python Version	3.3.0, Sep. 2012

Table 6.5: Anomaly detection - simulation environment

Customer-centric anomaly detection

Figure 6.20 shows an overview over the performed simulation runs for scenario User-anomaly03. Best performance regarding speed and learning rate are highlighted in green. During all simulations the target result was easily reached, thus a tighter target net error rate of 0.001 was chosen. The column *Simulation Time* represents the time of ten complete simulation runs of the corresponding network structure. It can be seen, that the ANN design influences the overall performance of the anomaly detection characteristics. Dependent on the requirements, the following can be derived from the results:

- Speed - If minimum learning speed is a desired quality, an ANN structure of 4-40-20-10-5-1 delivered the fastest learning speed. Ten complete simulation runs

Scenario: UserAnomaly03							
Simulation Parameters							
Test Repetitions		10					
DeltaCCT & DeltaCST		60 min					
Training duration		10 days					
No of abnormal VMs		20					
Lower Limit between two malicious VM-Create		30 min					
Upper limit between two malicious VM-DELETE		120min					
Normal VM Count		2					
Scenario	**ANN Parameters**		**Results**			**Meta Data**	
	ANN Structure	**Target Net Error**	**Learning Repetitions (Average)**	**Net Error after Learning Phase (Average)**	**Total Error Rate (Average) in %**	**Simulation Time in s**	**Result File**
UserAnomaly03	4-5-3-1	0.001	724	0	2,3	39	useranomaly03-result-evaluated-overall-4-5-3-1-2013-11-03-14-29-01.xls
UserAnomaly03	4-10-5-1	0.001	606	0	2,1	30	useranomaly03-result-evaluated-overall-4-10-5-1-2013-11-03-14-41-20.xls
UserAnomaly03	4-20-10-1	0.001	757	0	0	40	useranomaly03-result-evaluated-overall-4-20-10-1-2013-11-03-14-44-56.xls
UserAnomaly03	4-40-20-1	0.001	576	0	0	39	useranomaly03-result-evaluated-overall-4-40-20-1-2013-11-03-14-47-22.xls
UserAnomaly03	4-80-40-1	0.001	292	0	4,3	39	useranomaly03-result-evaluated-overall-4-80-40-1-2013-11-03-14-50-28.xls
UserAnomaly03	4-160-80-1	0.001	215	0	0	71	useranomaly03-result-evaluated-overall-4-160-80-1-2013-11-03-14-52-57.xls
UserAnomaly03	4-320-160-1	0.001	65	0	0	96	useranomaly03-result-evaluated-overall-4-320-160-1-2013-11-03-14-55-46.xls
	4-10-5-3-1	0.001	401	0	0	22	useranomaly03-result-evaluated-overall-4-10-5-3-1-2013-11-03-14-58-44.xls
UserAnomaly03	4-20-10-5-1	0.001	421	0	0	25	useranomaly03-result-evaluated-overall-4-20-10-5-1-2013-11-03-15-00-18.xls
UserAnomaly03	4-40-20-10-1	0.001	363	0	0	28	useranomaly03-result-evaluated-overall-4-40-20-10-1-2013-11-03-15-01-37.xls
UserAnomaly03	4-80-40-20-1	0.001	320	0	0	44	useranomaly03-result-evaluated-overall-4-80-40-20-1-2013-11-03-15-07-48.xls
UserAnomaly03	4-160-80-40-1	0.001	119	0	0	60	useranomaly03-result-evaluated-overall-4-160-80-40-1-2013-11-03-15-09-40.xls
UserAnomaly03	4-320-160-80-1	0.001	46	0	0	120	useranomaly03-result-evaluated-overall-4-320-160-80-1-2013-11-03-15-03-14.xls
UserAnomaly03	4-20-10-5-3-1	0.001	213	0	2	17	useranomaly03-result-evaluated-overall-4-20-10-5-3-1-2013-11-04-11-43-21.xls
UserAnomaly03	4-40-20-10-5-1	0.001	162	0	0	16	useranomaly03-result-evaluated-overall-4-40-20-10-5-1-2013-11-04-11-50-41.xls
UserAnomaly03	4-80-40-20-10-1	0.001	234	0	0	36	useranomaly03-result-evaluated-overall-4-80-40-20-10-1-2013-11-04-11-52-17.xls
UserAnomaly03	4-160-80-40-20-1	0.001	186	0	0	83	useranomaly03-result-evaluated-overall-160-80-40-20-1-2013-11-04-11-53-58.xls
UserAnomaly03	4-320-160-80-40-1	0.001	47	0	0	138	useranomaly03-result-evaluated-overall-320-160-80-40-1-2013-11-03-15-13-29.xls

Figure 6.20: Anomaly Detection - simulation results (UserAnomaly03)

were performed within 16 seconds, with a resulting detection rate of 98%.

- Learning Repetitions - If a minimum of learning repetition is aspired, an ANN structure of 4-320-160-80-1 performs best. The network only needed 46 epochs, with a resulting detection rate of 100%

Table 6.6 shows detailed results of ten simulation runs of scenario UserAnomaly03 with the ANN structure 4-40-20-10-5-1. It can be seen, that over ten simulation runs, the network was able to correctly identify normal and abnormal behaviour within the test data set of 10 events with an average detection error rate of 1.25%. The neural network needed in average 162 epochs to learn the behaviour of the simulated user to reach a net error rate of 0. The arithmetic average of a data row was calculated without the minimum and the maximum values.

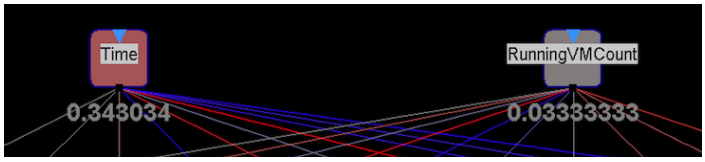


Figure 6.21: Anomaly detection - weighted input classifier (UserAnomaly01)

Run	Training Epochs	Net-Error	Total Test Patterns	Number of Anomalies	Correct Detected	Rows Mis-Detected	False Positives	False Negatives	Error Rate
1	224	0	10	5	10		0	0	0.0%
2	92	0	10	9	9	9	0	0	10.0%
3	329	0	10	5	10		0	0	0.0%
4	94	0	10	6	10		0	0	0.0%
5	103	0	10	8	10		0	0	0.0%
6	99	0	10	4	10		0	0	0.0%
7	71	0	10	5	10	8	0	0	10.0%
8	149	0	10	8	10		0	0	0.0%
9	394	0	10	10	10		0	0	0.0%
10	203	0	10	6	10		0	0	0.0%
⌀	162	0	10	6.5	9.875		0	0.125	1,25%

Table 6.6: Results of scenario Useranomaly03

For scenario UserAnomaly01 it can be seen in Figure 6.21 that the classifier "time" is used to determine if a CMS event is considered abnormal. This is also indicated by the weights and "trigger" indicators of the net after a simulation run. A deeper red background colour of a neuron indicates the influence of the neuron in the overall decision result. For scenario UserAnomaly02 and UserAnomaly03 the classifiers *LastVMCountCreate*" and *LastVMCountDelete* are mainly used for decision making, see Figure 6.22.

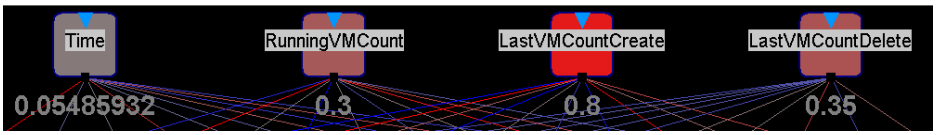


Figure 6.22: Anomaly detection - weighted input classifier (UserAnomaly03)

Cloud Wide Anomaly Detection

For the cloud wide anomaly detection, scenario ProviderAnomaly01 was simulated with different artificial network structures. Overall results are depicted in Figure 6.23. This time the usage behaviour of 50 users was simulated. For every user a "normal" VM-count between 0 and 5 was defined randomly, as well as corresponding core create and core stop times. Afterwards, one user was chosen randomly, and abnormal behaviour data for this user was added. Simulations were performed on the

Scenario ProviderAnomaly01							
Simulation Parameters							
Target Net Error	1						
Test Repetitions:	10						
Simulated user:	50						
Up to how many VMs p	5						
Deviation (in min.)	60						
Training period (in days	10						
Number of Abnormal Cl	30						
VM-OK Offset	0						

Scenario	ANN Parameters		Results			Meta Data	
	ANN Structure	Target Net Error	Learning Repetitions (Average)	Net Error after Learning Phase (Average)	Total Error Rate (Average) in %	Simulation Time in s	Result File
ProviderAnomaly01	6-10-5-1	1	269	0,5	5	124	provideranomaly01-result-evaluated-overall-10-5-1-2013-11-03-00-02-01.xls
ProviderAnomaly01	6-20-10-1	1	248	1,8	20	155	provideranomaly01-result-evaluated-overall-20-10-1-2013-11-03-00-02-01.xls
ProviderAnomaly01	6-40-20-1	1	106	0,8	10	147	provideranomaly01-result-evaluated-overall-40-20-1-2013-11-03-00-02-01.xls
ProviderAnomaly01	6-80-40-1	1	49	2,5	17	191	provideranomaly01-result-evaluated-overall-80-40-1-2013-11-03-00-05-47.xls
ProviderAnomaly01	6-160-80-1	1	17	4,4	8	257	provideranomaly01-result-evaluated-overall-160-80-1-2013-11-03-00-10-29.xls
ProviderAnomaly01	6-320-160-1	1	2	10,1	29	348	provideranomaly01-result-evaluated-overall-320-160-1-2013-11-03-00-16-15.xls
ProviderAnomaly01	6-10-5-3-1	1	319	0,75	9	130	provideranomaly01-result-evaluated-overall-6-10-5-3-1-2013-11-03-00-24-09.xls
ProviderAnomaly01	6-20-10-5-1	1	250	1,8	10	171	provideranomaly01-result-evaluated-overall-6-20-10-5-1-2013-11-03-00-28-19.xls
ProviderAnomaly01	6-40-20-10-1	1	128	3,5	18	195	provideranomaly01-result-evaluated-overall-6-40-20-10-1-2013-11-03-00-32-41.xls
ProviderAnomaly01	6-80-40-20-1	1	42	3,9	15	210	provideranomaly01-result-evaluated-overall-6-80-40-20-1-2013-11-03-00-37-38.xls
ProviderAnomaly01	6-160-80-40-1	1	13	6,8	20	240	provideranomaly01-result-evaluated-overall-160-80-40-1-2013-11-03-00-43-22.xls
ProviderAnomaly01	6-320-160-80-1	1	2	2,8	18	391	provideranomaly01-result-evaluated-overall-320-160-80-1-2013-11-03-00-48-51.xls
ProviderAnomaly01	6-20-10-5-3-1	1	454	2,6	32	264	provideranomaly01-result-evaluated-overall-6-20-10-5-3-1-2013-11-03-01-22-31.xls
ProviderAnomaly01	6-40-20-10-5-1	1	148	5,3	11	220	provideranomaly01-result-evaluated-overall-6-40-20-10-5-1-2013-11-03-01-17-22.xls
ProviderAnomaly01	6-80-40-20-10-1	1	46	4,6	19	220	provideranomaly01-result-evaluated-overall-80-40-20-10-1-2013-11-03-01-12-00.xls
ProviderAnomaly01	6-160-80-40-20-1	1	14	12,6	11	284	provideranomaly01-result-evaluated-overall-160-80-40-20-1-2013-11-03-01-05-40.xls
ProviderAnomaly01	6-320-160-80-40-1	1	2	4,5	25	437	provideranomaly01-result-evaluated-overall-320-160-80-40-1-2013-11-03-00-57-01.xls

Figure 6.23: Anomaly Detection - simulation results (ProviderAnomaly01)

same different neural network structures as for the customer-centric scenario, with a target net error of 1 during training phase. The following results can be stated from simulations:

- Speed - If minimum learning speed is a requirements, an ANN structure of 6-10-5-1 performs best. Ten complete simulation runs were performed within 124 seconds, with a resulting detection rate of 95%
- Learning Repetitions - If a minimum of learning repetition is aspired, an ANN structure of 6-160-80-1 performs best. The network only needed 17 epochs, with a resulting detection rate of 92%

Simulations show, that for this scenario, five out of 17 neural network structures meet the requirement of a maximum total detection error rate of 10%. Learning speed of the networks is overall slower than in the previous scenario. Both are due to a much higher number of events to learn from, including much more “noise” because of the highly differently acting users. Table 6.7 shows the results of ten simulation runs for the ANN structure 6-10-5-1. It can be seen, that over ten simulation runs, the network

Run	Training Epochs	Net-Error	Total Test Patterns	Number of Anomalies	Correct Detected	Rows Mis-Detected	False Positives	False Negatives	Error Rate
1	281	0	558	2	2		0	0	0.0%
2	177	1	626	6	5	333	1	0	25.0%
3	306	0	618	5	5		0	0	0.0%
4	112	6	550	6	5		0	0	0.0%
5	264	0	662	6	6		0	0	0.0%
6	255	0	634	4	4		0	0	0.0%
7	939	1	746	5	3	109, 165	2	0	40.0%
8	276	0	626	4	4		0	0	0.0%
9	276	1	618	4	3	516	1	0	16.7%
10	316	1	590	7	7		0	0	0.0%
⌀	267	0.5	616.5	5	616.25		0.25	0	5.2%

Table 6.7: Anomaly Detection - simulation results (ProviderAnomaly01)

was able to correctly identify normal and abnormal behaviour within the test data set of 616 events (average) with an detection error rate of 5,2%. The neural network needed in average 267 learning repetitions to learn the behaviour of the simulated users to reach the target net error. Again, the arithmetic average of a data row was calculated without the minimum and the maximum value. All simulation result files are available on the attached DVD in folder *CloudUsageSimulator/SimulationResults*. As seen by different simulation runs, the ANN's network structure and chosen features highly influence the performance of the system. An optimal result is dependent on the quality of the chosen input qualifiers [211]. Further optimisation criteria could be:

- Minimisation of training patterns needed to learn
- Improvement of learning speed: faster time to reach target net error
- Minimisation of overall detection error
- Identification of more or less important input classifier for specific scenario
- Reduction of learning period with same detection rate

However, due to respect of the whole work already achieved by the *Security Audit Compliance For Cloud Computing* research, and for time constraints on finishing the PhD thesis, this investigations are part of future work.

Although the previous section proved the feasibility of using neural networks as an anomaly detection system in cloud environments, discussion on some aspects of the research is necessary. Thus, this section continues to discuss the following aspects of the approach chosen:

- Neural network approach - Why neural networks are chosen?
- Usage of simulated data - Why real world data was not used?
- Training of impostor data - Is a practical deployment possible?

6.6 Discussions on the Neural Network Approach

A research area, closely related to anomaly detection in cloud environments are Intrusion Detection Systems. They share certain characteristics, such as a frequently changing infrastructure of networking components, huge networking data and a strong focus on detecting anomalies in usage of networking systems. Well known approaches for misusage detection or anomaly detection in IDS according to [215] are:

- Expert systems, containing a set of rules that describe attacks
- Signature verification, attacks are translated into sequences of audit events
- Petri nets, where known attacks are represented with graphical petri nets
- State transition diagrams, representing attacks with a set of goals and transitions
- Threshold detection detecting abnormal activity on a server or network, for example abnormal saturation of the network
- Statistical measures, learned from historical values
- Non-linear algorithms, such as Neural Networks or Genetic algorithms

An often used approach for anomaly detection is statistical analysis [215]. They are pervasively leveraged in industry monitoring products, such as HP Systems Insight

Manger [216], IBM Tivoli [217], Nagios [167] and even in IaaS cloud health monitoring systems, e.g., Ganglia [218]. Statistical approaches are outperforming neural networks in terms of speed by far, since the underlying calculation of anomalies are much faster. However, important for a successful statistical system is the definition of meaningful, well balanced thresholds. Determination of such is a difficult task, especially if behaviour changes slowly over time [211], [215]. Additionally, [219] also show, that threshold approaches do not work well in exascale, highly dynamic cloud environments.

By using neural networks, a determination of a threshold is not necessary, since the system learns the behaviour from the cloud usage as shown in Section 6.4.4. Furthermore, the use of neural networks also benefits from using actual event data rather than using parameters that are derived from the actual data (e.g. mean and standard deviation). While the examples presented above, only uses four input dimensions the system is able to be extended by any additional cloud metrics, such as VM utilisation (CPU usage, RAM usage, network traffic, process execution times, etc.).

Another advantage of using neural networks is, that they are well suited for anomaly detection due to their flexibility. They are able to analyse data, even from incomplete or distorted data sets [220]. Since their output is in form of a probability, it can be used as a predictive capability, which can be added to results of other / additional decision making parameters. In an SAaaS enable IaaS cloud, this could be, that the neural network evaluates a certain VM behaviour to be abnormal by 60%. In addition, the IP address gets added to the decision making process. Differs the IP range strongly from all prior seen IP ranges of legitimate access, the probability that the monitored behaviour is an anomaly is strongly increased. Furthermore, after validating if the monitored event was an actual anomaly, this information can be applied back to the learning process of the network.

This is also the most important advantage of neural networks. By combining multiple detection systems, such as the presented Rule based and Behavioural based detection method plus a possibility of manual feedback system by cloud provider's personnel or cloud user, the learning process of neural networks can be improved massively. Thus, the neural network can learn more characteristics of misusage cases.

Usage of Simulated Data

The presented approach utilises a cloud usage data simulator to artificially simulate cloud usage data sets. Compared to real world data it is limited, due to the fact, that a certain, modelled usage behaviour gets assumed in the simulation environment. Thus, no real-world validation of the proposed solution (e.g. by deploying it in a real infrastructure) but just simulations exists.

Before creation of the cloud usage simulator, the author of this research contacted multiple commercial cloud provider, to get some real cloud usage data. An email correspondence proving this is available in Appendix A.5. However, all contacted IaaS cloud provider refused to disclose parts of their cloud usage data. This is a phenomenon, which is not limited to this research. Orna et al. [221], TaheriMonfayed and Jaatun [222] or Phillips et al. [223] are just a three more examples of related research on cloud Computing security, which encountered the same problem. Cloud providers do not disclose any data. It is assumed that fear of industry espionage is a reason: "Conventionally, cloud providers are not willing to disclose details of their security mechanisms. They justify this behavior in different ways, but it seems that the main motivation is fear of competitors stealing their ideas" [222]. Furthermore, to the best knowledge of the author, no open source cloud usage data set was available during time of the research. The author of this research decided to utilise simulated cloud usage data to develop and test the presented anomaly detection system. The

system uses several input classifier of cloud usage data in an normalised form. But, now that the neural network based system exists, a possible way to test the system with real data is to give the cloud providers the neural networks to let them process their data with it. Thus, the performance of the developed system could be validated, without the necessity for cloud providers to disclose any data. However, this approach was not possible for this research, since the system needed to be developed first. Secondly, due to time constraints on finishing this PhD work, this task is up to future work.

Training of Impostor Data

Although *Chapter 6.4 - Behavioural Based Anomaly Detection* has proven the feasibility of utilising cloud usage behaviour profiles to train neural networks, a failing exist in the practical application of the system. Classification is performed by neural networks, which are trained using both, normal and abnormal data (impostor data). However, in a real world application, the availability and suitability of impostors will be limited, since the system does not provide any value if it needs to be compromised first to achieve full functionality. This is a well known problem, which also occurs in the area of bio metric authentication, discussed by Lecomte et al. in [224].
The current classification process of the anomaly detection system presented, utilises neural networks, where data of considered normal cloud usage is used alongside impostor data to teach the network the difference in cloud usage characteristics. Supervised learning is used to teach the network, which data belongs to the cloud user and which data belongs to an attacker or impostor. However, for a real world system, this approach is limited due the availability and performance of imposter data, for the following reasons:

- **Availability** - misusage cases of cloud instances (imposter data) will always be

required for the system to teach the neural network. This practically results in the requirement, that cloud instances needs to be compromised first, before the system can be used to detect anomalies.

- **Performance** - the same cloud usage data simulator is used, to generate the anomalies, the network is evaluated with. Although, the data samples are not used in training and testing of the network, this fact rises the possibility to skewed performance rates.

However, these limitations apply for the approach chosen by this work, research has already proven, how these limitation can be mitigated. Instead of using impostor data, Lecomte et al. has shown in [224] how impostor data can be artificially created from normal behaviour data. By adding a slight set of deviations to normal usage data, artificial imposter data is created. This eliminates the need of impostor data completely. Furthermore, their research shows, that training neural networks with artificially created impostor data outperforms usage of real impostor data by a 25% increased performance, leading to optimal configured classification engines.

6.7 Summary

This Chapter elaborated two approaches on mitigating cloud specific security challenges by anomaly detection. A rule based and a behavioural based detection system were presented by utilising several user-centric and cloud wide example scenarios. It was shown, that the developed SAaaS architecture in combination with the Cloud Audit Policy Language can be used for rule based anomaly detection. To prove the feasibility of behavioural based anomaly detection a cloud usage simulator got developed. A fully automatic simulation environment was presented, using neural networks to learn and evaluate cloud usage behaviour to detect anomalies.

7 Security Audit as a Service Prototype

"We don't trust it until we can see it and feel it."

(Unknown author)

This chapter presents a developed prototype of the Security Audit as a Service system, including all research work presented so far. The concepts of concurrent audits utilising agents gets presented, as well as a prototype of the Cloud Audit Policy Language and the anomaly detection system.

7.1 Introduction

To provide a prove of concept of the introduced Security Audit as a Service architecture (Chapter 4), the Cloud Audit Policy Language (Chapter 5) and the research on anomaly detection in IaaS Clouds (Chapter 6), a SAaaS prototype has been developed. It is build upon the Cloud Infrastructure and Applications (CloudIA), which gets introduced at the beginning of this chapter. Then, a first prototype of the agent based cloud audit system SAaaS gets presented. Details on the utilised agent architecture and technical details on an agent's design are given. A demonstration (SAaaS Demo 1) shows a use case how SAaaS agents are used to detect changes on a cloud instance and perform an immediate cloud audit. The chapter then continues with the presentation of the Cloud Audit Policy Language prototype implementation. The CAPL server and its corresponding GUI frontend are elaborated in detail and it is shown how the SAaaS prototype is extended by it. A second demonstration (SAaaS Demo 2) presents how security audit policies can mitigates the cloud specific risk of scalability attacks. In the last section of this Chapter, the integration of the introduced anomaly detection system into the SAaaS prototype gets presented. Details on the implementation of a cloud usage data simulator are given. As a last stage of expansion, SAaaS Demo 3 shows, how the SAaaS prototype is extended by an anomaly detection system to detect user concentric and cloud wide anomalies in cloud usage behaviour.

Parts of this research phase have been published throughout all released papers of this research. Thus, they are not listed here in detail. A complete list of publication of this research is available at the end of this thesis as Chapter *"List of Publications"*.

7.2 The CloudIA Project at Furtwangen University

To harness the potentials of cloud computing for research at the Furtwangen University, and to SMEs, the Cloud Infrastructure and Applications project was established in 2009. The main objective of this project is to build a private cloud for the purpose of running e-Science and e-Learning applications at Furtwangen University. The overview of the CloudIA architecture is shown in Figure 7.1.

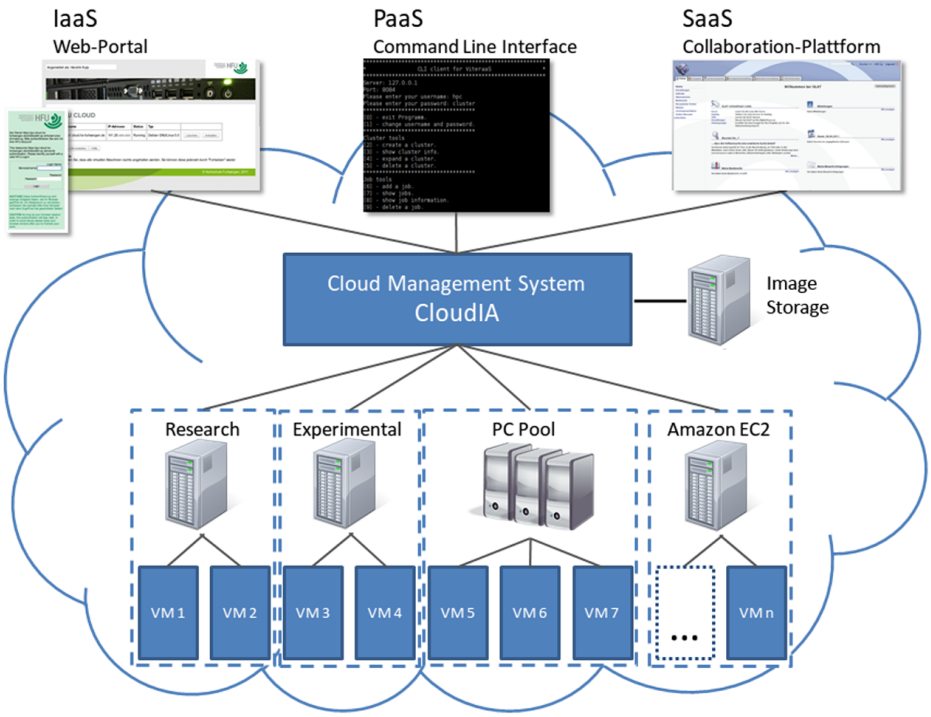


Figure 7.1: Cloud Infrastructure and Applications architecture [13]

The cloud architecture is build on top of an existing hardware infrastructure at Furtwangen University. It consists of three computer pools (PC Pool, Research Pool and Server Pool), that are located at different locations and use separate IP subdomains within the university. In addition, public cloud provider environments, such as Amazon EC2 are utilised as well to develop interoperable cloud solutions. CloudIA

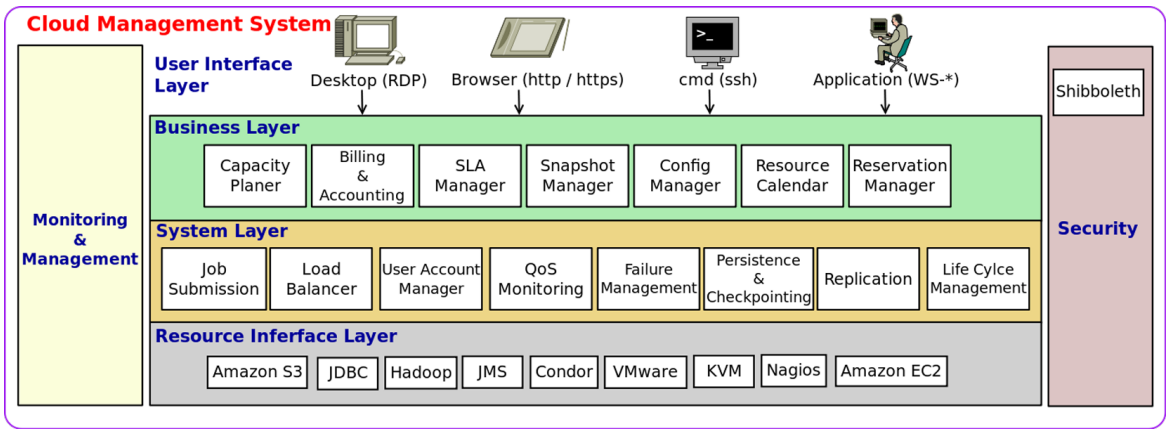


Figure 7.2: Cloud Infrastructure and Applications architecture module overview

leverages various virtualization technologies, such as Xen and KVM, and supports Service-Level Agreements (SLAs) for IaaS and SaaS models, as shown in Figure 7.2. In this figure, the cloud management system of CloudIA is divided into several layers for extensibility and maintainability. The following provides a brief introduction into the architecture.

User Interface Layer - This layer provides various access points to users and / or an administrator of the CMS in accessing our cloud system.

Business Layer - This layer aims to regulate resource supply and demand through the use of economy and SLA. In addition, this layer enables users to reserve VMs in advance and manage their personal VMs.

System Layer - This layer is responsible for daily operation of the CMS, such as submitting jobs, managing user accounts and monitoring Quality of Services (QoS) attributes.

Resource Interface Layer - This layer deals with the physical hardware and hosts many interfaces and plugins to various virtualization, database, distributed system and other technologies, such as Xen, Amazon EC2 and S3, and Nagios [167].

Monitoring & Management Component - To ensure reliability of each layer in the system, monitoring and management facilities are provided, specifically designed

for cloud administrators.

Security Component - To ensure privacy, recovery, integrity and security of user data and transactions, a security feature on all layers is required. Besides the technical solutions, issues in areas such as regulatory compliance and data auditing are important.

Designing the CloudIA architecture, was the first task of the research, presented in this thesis. It builds the architectural and technological base of every developed system at the Cloud Research Lab at Furtwangen University. Up to today, five granted research projects funded by the European Union or the German Federal Ministry of Education and Research (BMBF), listed in Table 7.1 are using it as a development base for research on cloud technologies.

Project Name	**Funded by**	**Duration**	**Description**
Security Audit as a Service (SAaaS) [225]	German Federal Ministry of Education and Research	2 years	Enhancing cloud security by cloud audits
StudiCloud [226]	Furtwangen University	unlimited	Provide university wide IaaS cloud infrastructure for Furtwangen University
Accountability for cloud and Other Future Internet Services (A4Cloud) [227]	European Union, 7th Framework Programme	4 years	Increase trust in cloud computing by accountability measures
Autonomic SLA Management as a Service for Cloud Services (ASLAMaaS) [228]	German Federal Ministry of Education and Research	3 years	Independent and automated management of Service Level Agreements in cloud environments
Ambient Assisted Living: Person Centred Environment for Information, Communication and Learning (AAL-PCEICL) [229]	Ministry of Sciences, Research and Arts Baden Wuerttemberg	3 years	Ambient assisted technologies for supporting elderly people

Table 7.1: Research projects build on CloudIA architecture

CloudIA Hardware

To provide a secure development process for CloudIA, three different cloud infrastructures were setup, as depicted in Figure 7.3:

- Experimental Cloud
- Research Cloud
- Productive Cloud

All cloud environments are independent from each other, operated by an own Cloud Management System. The **Experimental Cloud** consists of four AMD Athlon 64X2 Dual Core PCs with 2GB RAM each. It uses KVM as a hypervisor software for virtualization and runs Ubuntu Linux 12.04 LTS. It is mainly used for development of completely new cloud software, done by Bachelor or Master thesis students. It is further used as a staging cloud for the University's wide productive cloud and the Research Cloud, trying new software components or OS or hypervisor updates.
The **Research Cloud** environment is used for the development of research software, mostly by PhD students and research project personnel. It consists of five Intel Xeon PCs, each containing two Intel Quad Core 2.80 GHz processors with 12GB of RAM each. It uses KVM as hypervisor software and Debian 6.0 as Operating System. The Research Cloud is the technical base of the Security Audit as a Service prototype.
The **Productive Cloud** runs the Universities wide IaaS cloud environment StudiCloud [226]. It consists of six Intel Xeon machines, each containing 2 Intel Xeon QuadCore CPUs operated at 3.06 GHz and 64GB RAM each. It uses KVM as hypervisor software and Debian 6.0 as operating system. It is used by HFU's personnel and students and hosts in average 100 running VMs all the time. For cloud Bursting scenarios a connection to Amazon EC2 cloud exists as well. It is mainly used by the Research cloud to develop and demonstrate provider interoperable cloud software.

Storage is realised as a network attached storage running an ext4 filesystem. It gets accessed by the VMs via the Network File System (NFS) protocol, version 4. For VM images, base images [230] are used in combination of copy on write functionality. For every VM image available for deployment in one of the three Clouds a base images exists, containing an installed operating system. This is called a VM template. In case a User creates a new VM, it will always build on top if that VM template, called copy on write image. It is based on the base image, thus the image size barely contains any data when booted up for the first time. Thus, a fast deployment of new VMs is provided. Furthermore storage space is saved, since multiple copies of operating systems installed in different VM images are avoided.

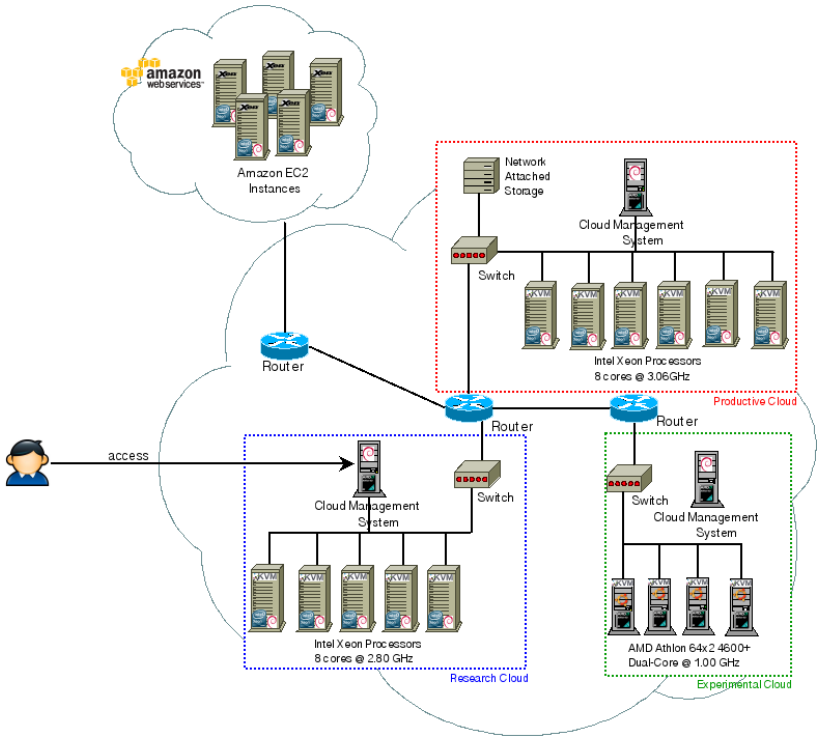
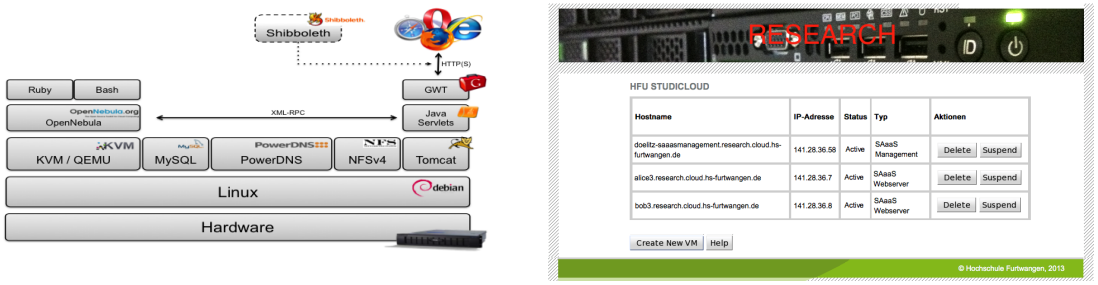


Figure 7.3: CloudIA hardware infrastructure at Furtwangen University



(a) CloudIA software stack (b) CloudIA web frontend GecMiGUI

Figure 7.4: CloudIA software components

CloudIA Software

Figure 7.4(a) shows the software stack, deployed on each of the three CloudIA infrastructures cloud Management Systems. On top of the hardware, Debian [231] Linux is used as an operating system (Ubuntu for Experimental Cloud). KVM is used as a hypervisor software, partitioning the real existing hardware into logical, separable parts. OpenNebula [159] is used as a cloud Management System, orchestrating the physical and virtual resources. Information about cloud hosts, VMs and peripheral components are stored in a MySQL database. On top of that, custom cloud software projects are developed, such as StudiCloud software. It interacts with the CMS and provides additional cloud management functionality, like the backend for the HFU's cloud Frontend (GecMiGUI). The front end is implemented as a web application, utilising the Google Web Toolkit (GWT) [232]. It is running in an exclusive VM on the same host as the Cloud Management System. Apache Tomcat [233] is used as an application server. The GUI provides basic VM management functionality such as a wizard based VM Create dialog and the possibility to resume or delete a VM. Figure 7.4(b) shows a screen shot of the Research cloud web interface with three running VMs. The cloud web interface is secured by a Single Sign-On system Shibboleth. The whole CloudIA software stack is released as Open Source software and available on Source Forge for download under the project name "StudiCloud" [234].

SAaaS Prototype Development

The SAaaS prototype is running on the HFU's Research Cloud. During the research, the prototype development went through the following different stages, which are described in the following in more detail.

1. **Development of SAaaS agents and SAaaS Management Interface - Section 7.3**
   - Research on available agent frameworks
   - Implementation of agent framework
   - Development of agents
   - Development of SAaaS GUI
   - **Result:** SAaaS Demo 1 - Detection of cloud infrastructure changes
2. **Extension of SAaaS prototype by a cloud audit policy language - Section 7.4**
   - Development of CAPL language definition
   - Implementation & integration of CAPL server into SAaaS prototype
   - Extension of SAaaS GUI by Policy Modeller
   - **Result:** SAaaS Demo 2 - Mitigating scalability attacks in IaaS clouds
3. **Extension of SAaaS prototype by an anomaly detection system - Section 7.5**
   - Development of cloud usage simulator
   - Development of neural networks for cloud behaviour analysis
   - Extension of SAaaS prototype by anomaly detection
   - **Result:** SAaaS Demo 3 - Anomaly detection in IaaS clouds

Based on this development process, the developed SAaaS prototype gets presented within the next sections.

7.3 SAaaS Prototype 1 - Integration of an Agent framework

The first version of the SAaaS prototype includes the deployment of an agent framework to an IaaS cloud, the development of SAaaS agents and a demonstration to detect changes within a cloud infrastructure.

7.3.1 Agent Development

As introduced in *Chapter 4.6 - Java Agent Development Framework*, JADE is used as an underlying agent framework for the SAaaS architecture. It offers the following default agents for agent management tasks:

- **Agent Management System (AMS)** - Agent, responsible for any inter-agent or container communication. Furthermore, the agent is responsible for creation and deletion of other agents within the same JADE platform
- **Directory Facility (DF)** - Agent, which offers agent management services, such as yellow page searches for agents. This service was not used within the SAaaS architecture, due to performance problems.

In addition, to the JADE default agents, the following Management Agents get deployed to every agent platform of SAaaS-enabled resources:

- **Agent Mobility Manager (AMM)** - Manages the mobility of an agent between the different agent platforms
- **Socket Proxy Agent (SPA)** - Enables any application to communicate with a SAaaS agent through a socket connection

- **Event Aggregator Agent** - Collects and pre-analysis events of sensor agents, before forwarding them to the Manager Agent

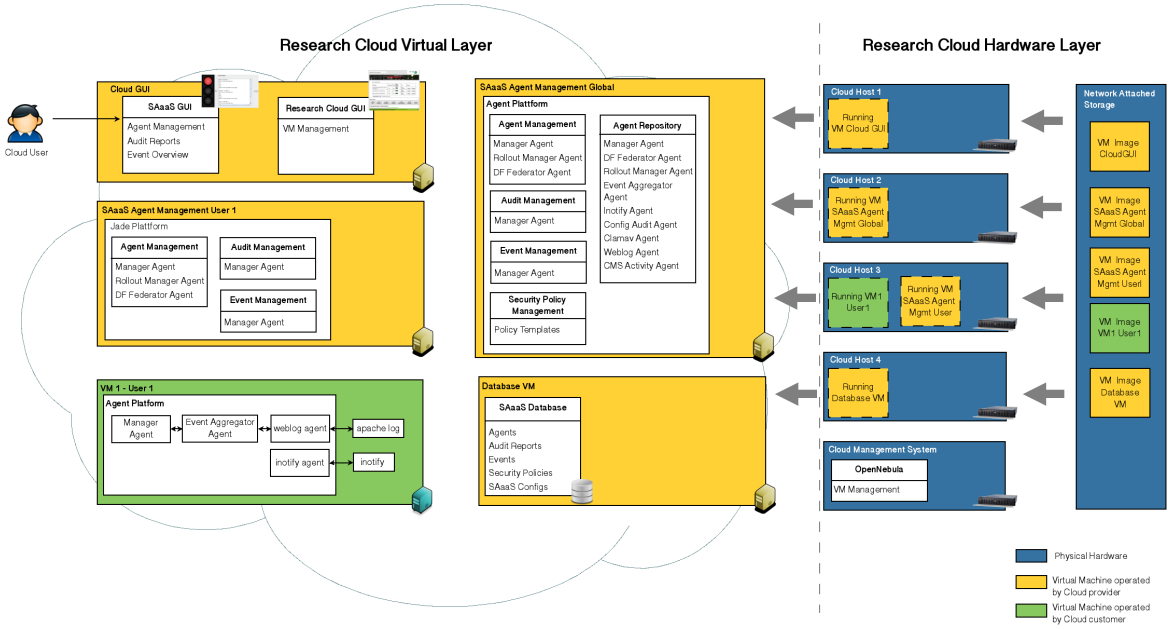


Figure 7.5: SAaaS prototype components - version 1

Figure 7.5 shows an overall view of all SAaaS prototype components. It shows the cloud storage, which stores all VM images of all users as well as images of management VMs[1]. Based on those images, the corresponding virtual machines are running, distributed over several physical cloud hosts[2]. A global SAaaS Agent Management system is running on an exclusive VM. The same applies for the SAaaS database VM and the cloud GUI, hosting the Research cloud web GUI and the SAaaS GUI. The components, depicted in Figure 7.5 assume, that a cloud user has already created one VM (VM1-User1) and enabled the SAaaS prototype features. Hence, automatically in addition to his VM, a user specific SAaaS Agent Management VM (SAaaS Agent Management VM User1) was started as well. This is done to keep the system flexible

[1] In an enterprise environment, storage space is distributed and separated between user images and provider images

[2] The distribution in this Figure is randomly chosen. Again, in a real enterprise scenario segmentation between user and provider VMs would be necessary

and scalable in respect to huge cloud infrastructures. It is furthermore assumed, that the user already configured a security policy. For the SAaaS prototype version 1 this is provided through prepared agent templates, available in the SAaaS graphical web interface. It was defined, that the configuration of the web server, running on the VM is finished. For the prototype, this results in the following automated actions: An inotify agent gets created from the SAaaS agent repository and configured to monitor the configuration file of the web server. Secondly, a weblog agent gets deployed monitoring the content of the web server's document root and the web server's logfile. Both sensor agents were deployed automatically to the user's VM. Figure 7.6 shows a cutout of the agent platform's logfile at the VM of user 1.

```
17.09.2013 15:05:48 de.hfu.wolke.saaas.caplAgents.InotifyAgentCAPL afterMove
INFO: Inotify Agent -- ready for CAPL arrived, going to work now
17.09.2013 15:05:49 de.hfu.wolke.saaas.caplAgents.InotifyAgentCAPL afterMove
INFO: Monitoring for /etc/apache2/sites-enabled/000-default with Events: modify added
17.09.2013 15:05:49 de.hfu.wolke.saaas.demo1.EventAggregatorAgent afterMove
INFO: Event Aggregator Agent arrived, going to work now
```

Figure 7.6: SAaaS prototype version 1 - Agents deployed

Table 7.2 shows a list of agents, which were developed during this stage of the SAaaS prototype. A comprehensive list of all developed agents during this research including technical details is provided in Appendix A.2.

7.3.2 SAaaS Graphical User Interface

For agent management tasks and to display events and corresponding audit reports, a graphical user interface is developed. It is hosted on a separate VM, running its own JADE platform for communication to SAaaS agents, see Figure 7.7. Agent configuration and events are stored in a SAaaS database, which is managed by the SAaaS Agent Management Platform. Agent communication is implemented utilising the ACL language format, introduced in *Chapter 4.6 - Java Agent Development Framework*.

Agent name	Agent Type	Position	Description
Manager Agent	Management agent	SAaaS-Agent Management Platform (global & user specific)	Management of agent events, delivers audit report events to SAaaS GUI, starts audit agent in case received event requires audit
Rollout Manager Agent	Management agent	SAaaS-Agent Management Platform (global & user specific)	Creates an agent, which then deploys itself via *doMove()* to a target agent platform at a target VM or cloud host
DF Federator Agent	Management agent	SAaaS-Agent Management Platform (global & user specific)	Enables federation of agents to agent yellow pages
Event Aggregator Agent	Management agent	Target VM / cloud host	Collects and pre-analysis events of sensor agents, sends events to Manager Agent
Socket Proxy Agent	Management agent	Target VM / cloud hosts	Enables message communication between non-agent applications (agent plugins) and a SAaaS agent
CMS Activity Agent	Sensor agent	Cloud Management System	Monitors cloud Management System events, such as VM-CREATE or VM-DELETE commands
Inotify Agent	Sensor agent	To monitored VM / cloud host	Monitors filesystem changes, reports to Aggregator Agent
Config Audit Agent	Sensor agent	Target VM / cloud host	Gets deployed to a target VM or cloud host. Contains file list, which it parses for necessary or forbidden key words
ClamAv Agent	Sensor agent	Target VM / cloud host	Gets deployed to a target VM or cloud host. Contains file list, which gets scanned for viruses or malware
Weblog Agent	Sensor agent	Target VM / cloud host	Gets deployed to a target VM or cloud host. Monitors web server log file for file access

Table 7.2: SAaaS agents developed during 1st SAaaS prototype stage

Transportation of messages is done via the HTTP protocol, using a Trusted Layer Security (TLS) encrypted connection. Figure 7.7 shows a typical setup of the SAaaS prototype version 1, with the deployed default agents described above. Furthermore, it shows a cloud customer VM with enabled SAaaS features. Thus an agent platform is running on it, and one agent is already deployed on it. This agent is fed from two agent plugin tools. One is able to communicate directly with the agent, the other is not compatible to a JADE agent, thus it is communicating through the Socket Proxy Agent to the SAaaS system. Figure 7.8 shows the Cloud Management GUI extended by elements of the SAaaS prototype, highlighted in green. For version 1 of the SAaaS prototype, the functions *Enable SAaaS Features*, *SAaaS VM Agent GUI*, *SAaaS Secu-*

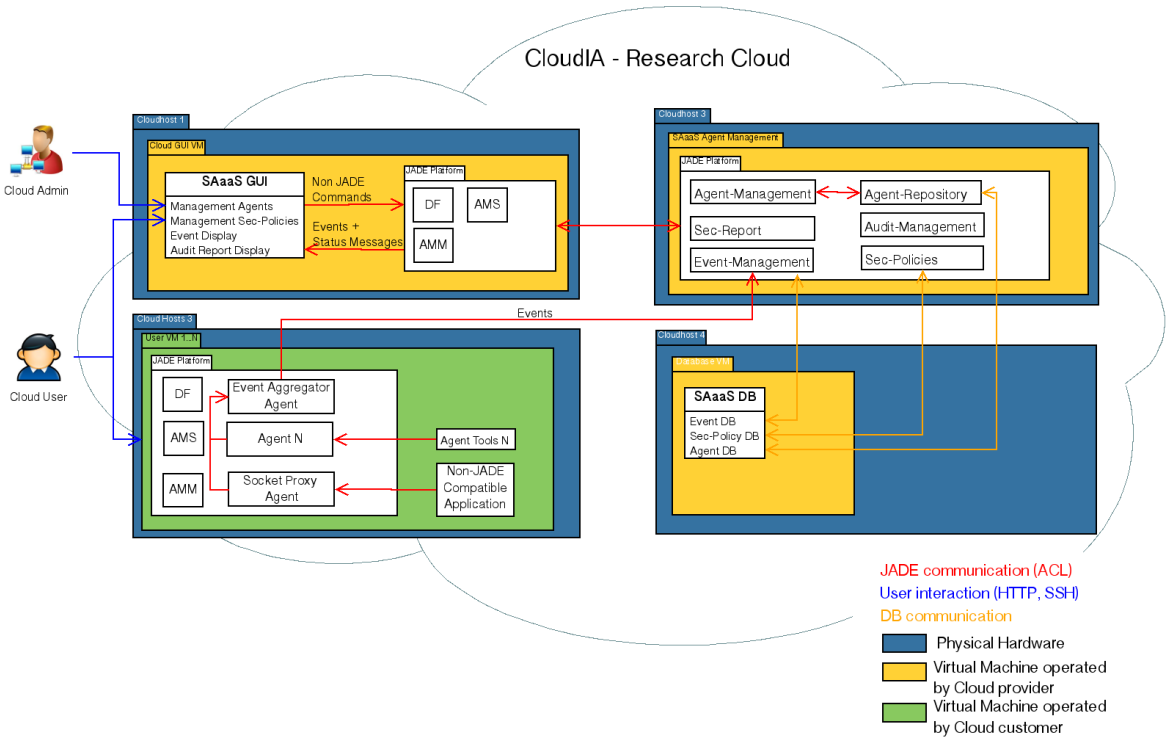


Figure 7.7: SAaaS prototype - SAaaS agents

rity Dashboard, *SAaaS Audit Report Dashboard* and *SAaaS Provider Dashboard* were implemented. The other functions are added in later versions of the prototype.

1. SAaaS Agent VM GUI - allows the management of agents, see Figure 7.9(a). It shows a simple three colour signal indicator it the JADE platform on the corresponding cloud resource is available and reachable. It allows the deployment of prepared security policy templates or agent configurations. Agents running on a cloud resource can be graphically displayed via the button `Show running agents`, its content is depicted in Figure 7.9(b). In this case, the default SAaaS management agents (listed in grey) as well as an inotify and the Event Aggregator agents are running on the target VM "webserver1".

2. SAaaS Security Dashboard - submitted agent events are visualised in the SAaaS Security Dashboard, depicted in Figure 7.10(a). It shows cloud customer specific events, about detected events among his cloud instances. A simple security in-

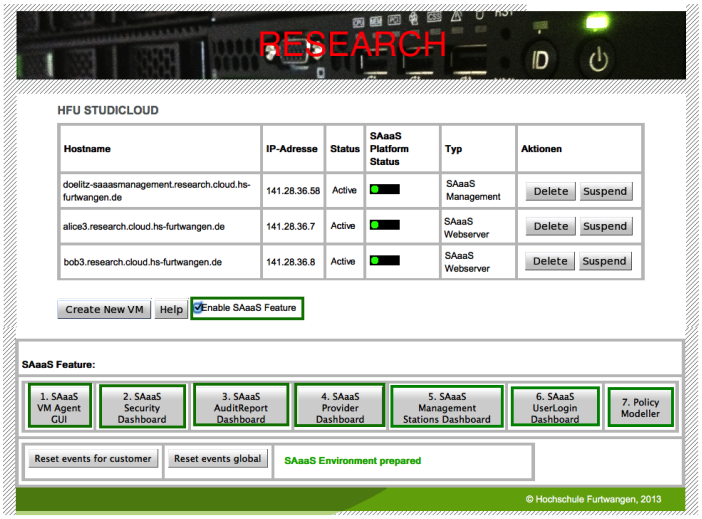
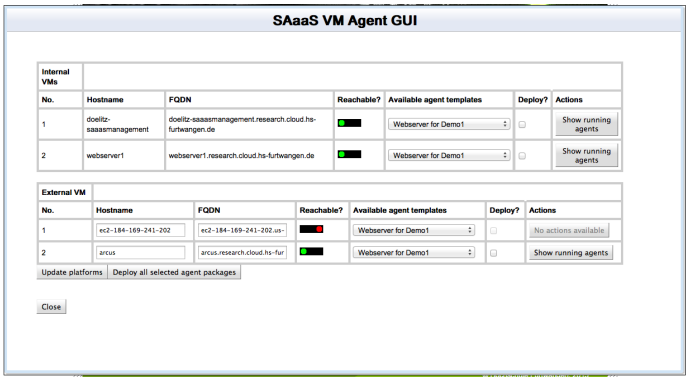
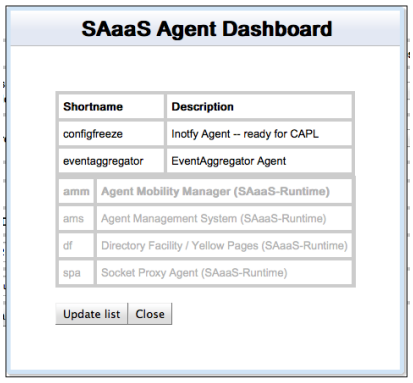


Figure 7.8: SAaaS prototype: SAaaS GUI elements



(a) SAaaS prototype: SAaaS Agent VM GUI



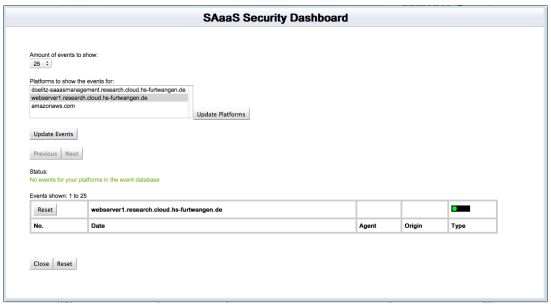
(b) SAaaS prototype: Running agents on SAaaS enabled VM

Figure 7.9: SAaaS prototype: graphical agent management

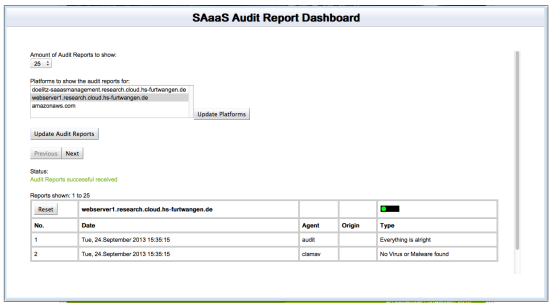
dicator signal gives a quick overview over the security status of the deployed cloud instances.

3. SAaaS Audit Report Dashboard - audit reports are displayed in the SAaaS Audit Report Dashboard, depicted in Figure 7.10(b). Again, a simple signal indicator light informs about the overall status of audit reports.

4. SAaaS Provider Dashboard - Since the provider might be interested in cloud customer overspanning events, the SAaaS Provider Dashboard shows events of all



(a) SAaaS prototype: Security Dashboard



(b) SAaaS prototype: Audit Reports Dashboard

Figure 7.10: SAaaS prototype: Security & Audit Dashboard

SAaaS enabled resources, as depicted in Figure 7.11. The administrator can choose for which cloud resource he wants to list events. For debugging, all raw events produced by Sensor or Metric Agents are displayed (Figure 7.11 column: "Events collected by the Event Aggregator"), as well as the corresponding events send from the Event Aggregator Agent to the corresponding Manager Agent (Figure 7.11, column "Events send to the Manager Agent").

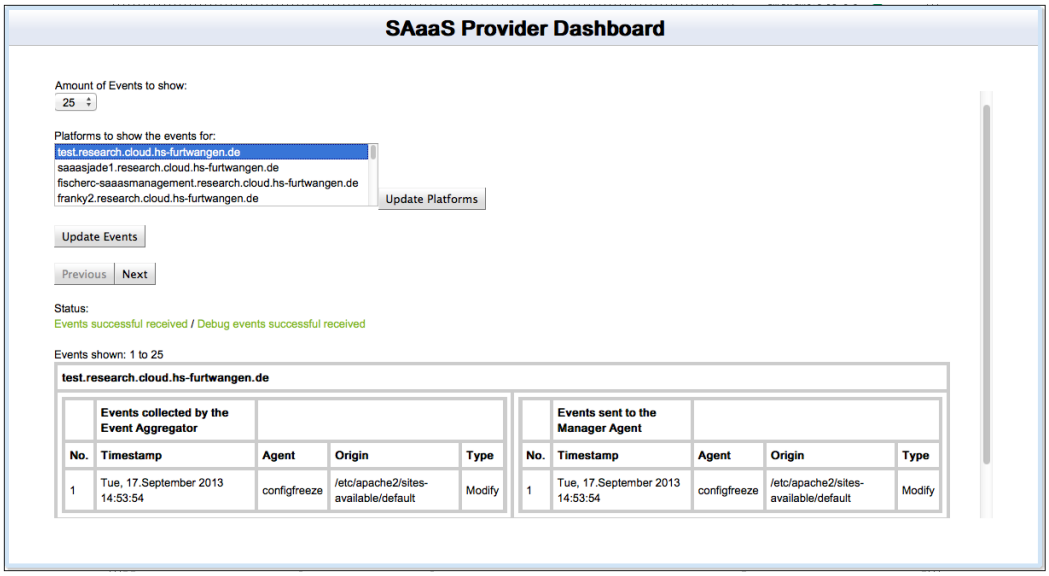


Figure 7.11: SAaaS prototype: SAaaS Provider Dashboard

7.3.3 SAaaS Demo 1 - Detection of Cloud Infrastructure Changes

To demonstrate the developed SAaaS architecture, the following two demos are presented:

- Demo 1a - Detection of an attacked VM
- Demo 1b - Detection of a security misconfiguration

Demo 1a - Detection of an Attached VM

In the first SAaaS demo the functionality of the agent framework, and its successful integration into the CloudIA cloud infrastructure is demonstrated. The following attack scenario is addressed: Detection of a change in a cloud infrastructure. An attacker is able to successfully login to a target VM via SSH. Then, multiple malicious files (e.g. a trojan) get uploaded, one amongst others to the webserver's directory. This file then gets downloaded by a possible future victim. The chronological sequence of the demo is shown in Figure 7.12. It contains three major phases:

1. **VM preparation** - Deploy of a webserver VM and assignment of a SAaaS security policy template "web server protection"
2. **Attack & detection** - VM gets attacked, agents generate multiple simple events
3. **Evaluation** - Attack is reported to SAaaS dashboard

1. VM preparation. First a cloud user deploys a new VM and configures it as a webserver (Figure 7.12, ①). After finishing the configuration of the VM, he enables the SAaaS features within the SAaaS web GUI. As a result, automatically a SAaaS agent management VM gets started, containing the default agents. The user then assigns a predefined security policy template "webserver-apache2" to it ②, as shown in Figure 7.13. The template contains three security policies and corresponding agent configurations:

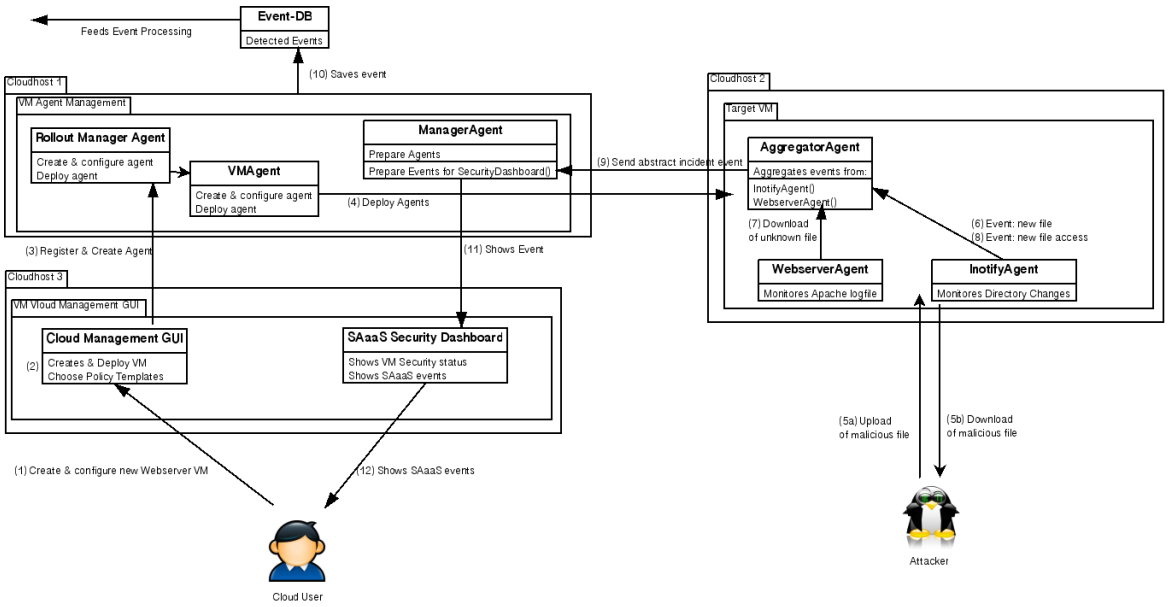


Figure 7.12: SAaaS Demo 1 - Detection of cloud infrastructure changes

- Monitoring of Apache web server config directory
- Monitoring of web server document root directory
- Allowed authentication for Apache web sites: `HTTP-DIGEST`

By enabling those security policies, this automatically prepares ③ and deploys ④ an inotify Agent and a Weblog Agent to the running VM. The inotify Agent is monitoring changes to the webservers' config and webroot directory, whereas the Weblog Agent compares http-get requests with a list of allowed html files of the webroot directory. Additionally, an Event Aggregator Agent gets deployed to the target VM. It includes a simple message reduction method for the Aggregator Agent stating: "All simple events occurring within one minute by these two agents get combined into one "Webserver attacked - unknown file detected" message".

2. Attack & detection. An attacker logs in to the VM, uploads a malicious file to the VMs webserver root ⑤a directory. This then gets downloaded ⑤b by the attacker to check the success of his attack. This automatically results in three simple events generated by the corresponding agents:

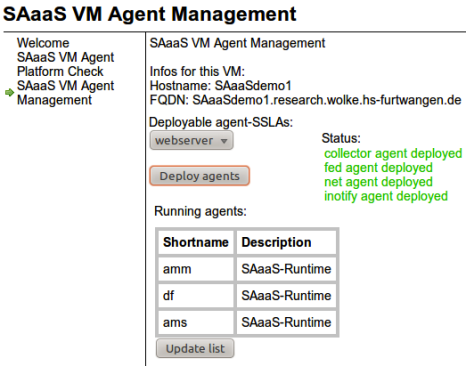


Figure 7.13: Assignment of security policy template for web server protection

Inotify Agent: creation of new file in webserver-root directory ⑥, read access of new file ⑧ *Weblog Agent*: download of unknown file ⑦.

These three simple events get combined to one abstracted "Webserver attacked - unknown file detected" event by the Aggregator agent due to the simple message reduction method mentioned above, and sent out ⑨ to the Manager Agent module at the user's Agent Management Platform.

User notification. All events get saved into the user specific event database and forwarded to SAaaS' global event management module ⑩. By doing this, cloud wide incident detection can be achieved, described later in Section 7.5. The user gets informed ⑪ via the SAaaS Security Dashboard, depicted in Figure 7.14. It shows the VM's security state before an attack. After launching an attack, the Security Dashboard indicator light changes its colour as set defined in a simple severity matrix and gives short information about the monitored event (Figure 7.14).

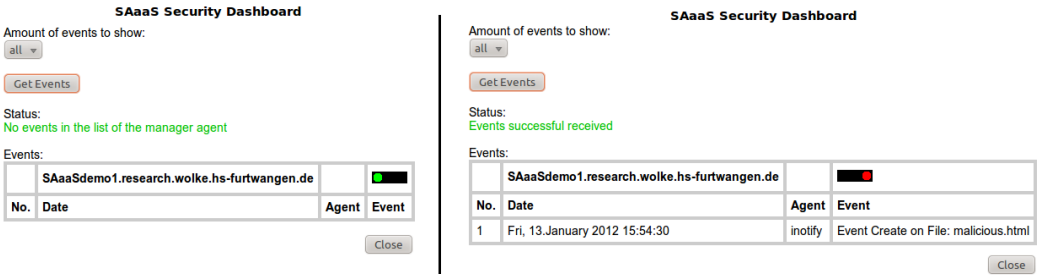
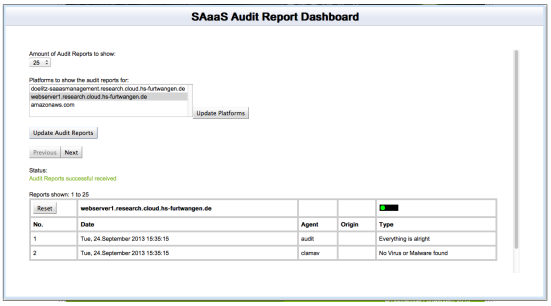
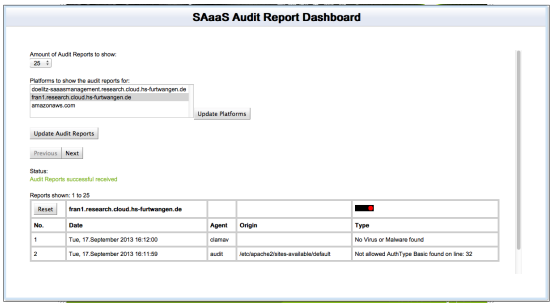


Figure 7.14: Cloud security dashboard prototype before and after detected attack



(a) SAaaS prototype: Demo 1b Audit report before infrastructure change



(b) SAaaS prototype: Demo 1b - Audit report after infrastructure change

Figure 7.15: SAaaS prototype: Demo 1b - concurrent audits

Demo 1b - Concurrent Audit Due to Cloud Infrastructure Change

In demo 1b, the concept of concurrent audits, as a reaction of a detected cloud infrastructure change is shown. Again, the demo is divided into three parts:

1. **VM preparation** - A new VM gets created
2. **Detection** - A change gets detected
3. **Evaluation & reporting** - The change gets analysed, evaluated and reported

1. VM preparation. This scenario assumes, that a cloud user is already running some SAaaS-enabled web server VMs within an IaaS cloud. As a general security policy it is defined, that only an encrypted `HTTP-DIGEST` authentication is allowed for web server content. On a regular base, an Audit agent gets deployed to all VMs, checking the security status of those VMs, including the configuration of the web server. At the beginning of this demo, the SAaaS Audit Report Dashboard shows the results of the last audit, depicted in Figure 7.15(a).
Now, the cloud user creates a new web server to the Cloud. He configures it, but accidentally enables the unencrypted `HTTP-BASIC` authentication. He finishes the VM configuration and closes the connection to the newly created VM.

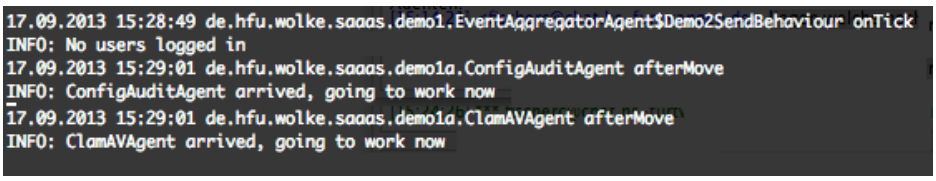


Figure 7.16: SAaaS prototype: Demo 1b Audit agent arrived at target VM

2. Detection. The CMS Agent monitoring cloud Management actions, such as the creation of new VMs, detects the newly created VM of the cloud user. This is considered a cloud infrastructure change. The information gets forwarded to the Manager Agent at the user's SAaaS Management Platform, including the type of VM (web server). Automatically, all existing security policies for VMs of the type "web server" get loaded and an Audit agent gets configured. As soon as the user logs out of the newly created VM, the Audit agent gets deployed to this VM, performing a compliance check of all security policies. Figure 7.16 shows the logfile of the JADE platform at the newly created VM. It states, that a new audit agent is arrived, starting its audit checks. During this check, the misconfigured web server configuration (HTTP BASIC AUTH) gets detected.

3. Evaluation & Reporting. Results of the audit are reported back to the Manager Agent at the user's Agent Management Platform. The audit gets saved into the user's audit report database, and the results get displayed within the SAaaS Audit Report Dashboard. Figure 7.15(b) shows the SAaaS Audit Report Dashboard with the detected VM misconfiguration.

To show, that the developed SAaaS architecture is compatible to other cloud providers, the newly created web server VM was created within the Amazon EC2 Cloud, whereas the SAaaS Agent Management Platform and the other (pre-existing) web server VMs are running within the CloudIA infrastructure. This demo was presented at the 8th IEEE World Congress on Services 2012 - IEEE Services CUP and was awarded with the second prize [235]. The conference poster can be found in Appendix A.7, the corre-

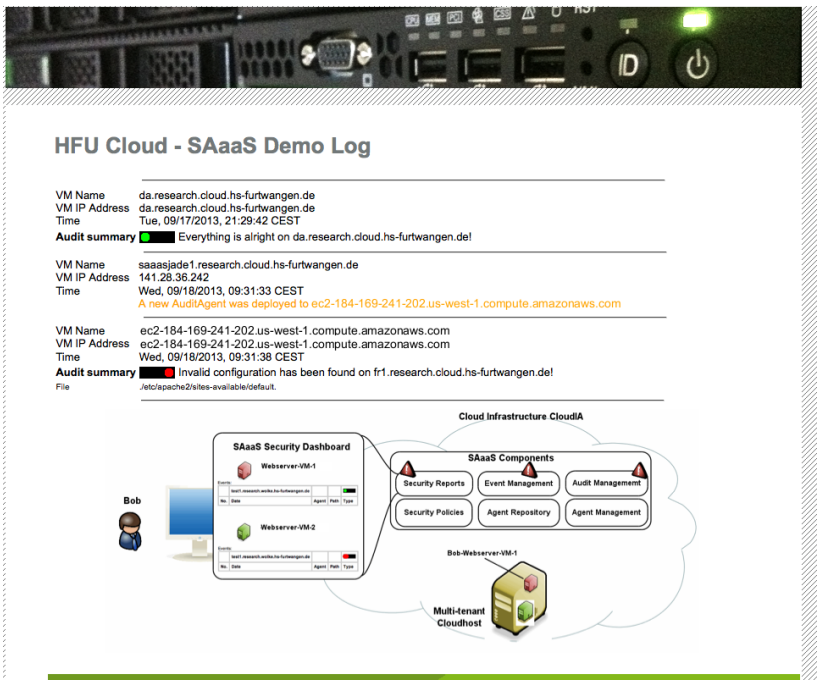


Figure 7.17: SAaaS prototype: Demo 1b graphical log of demo sequence

sponding paper "Validating cloud Infrastructure Changes by Cloud Audits" [144] can be found on the attached DVD. For a better understanding of the demo sequence, a graphical log system was developed, showing the sequence of events recognised by the SAaaS Manager Agent during the demo. It is depicted in Figure 7.17. It shows the results of the first audit, at an existing VM "da.research.cloud.hs- furtwangen.de". Afterwards the event about an automatically deployed Audit Agent to an Amazon EC2 VM is shown, followed by a detailed message on the found security misconfiguration. To visualise the dynamics of the system, an animated movie showing the sequential agent creation was done. It is available on the DVD attached to this thesis in the folder *SAaaS_Prototype_Demos/SAaaSDemo1bConcurrentAudit.mov* or online at http:// doeli.de/web/video/SAaaSDemo1bConcurrentAudit.mov.

7.4 SAaaS Prototype 2 - Integration of the Cloud Audit Policy Language

As a second step, the Cloud Audit Policy Language, presented in Chapter 5 is seamlessly integrated into the SAaaS architecture. Whereas security policies were predefined as templates in the SAaaS prototype version 1, they will be modelled through a graphical policy modeller and managed by a policy server. Thus, the prototype was extended by a CAPL server for policy management and a graphical policy modeller. Figure 7.18 shows the SAaaS prototype version 2 architecture, extended by the CAPL components (highlighted in green).

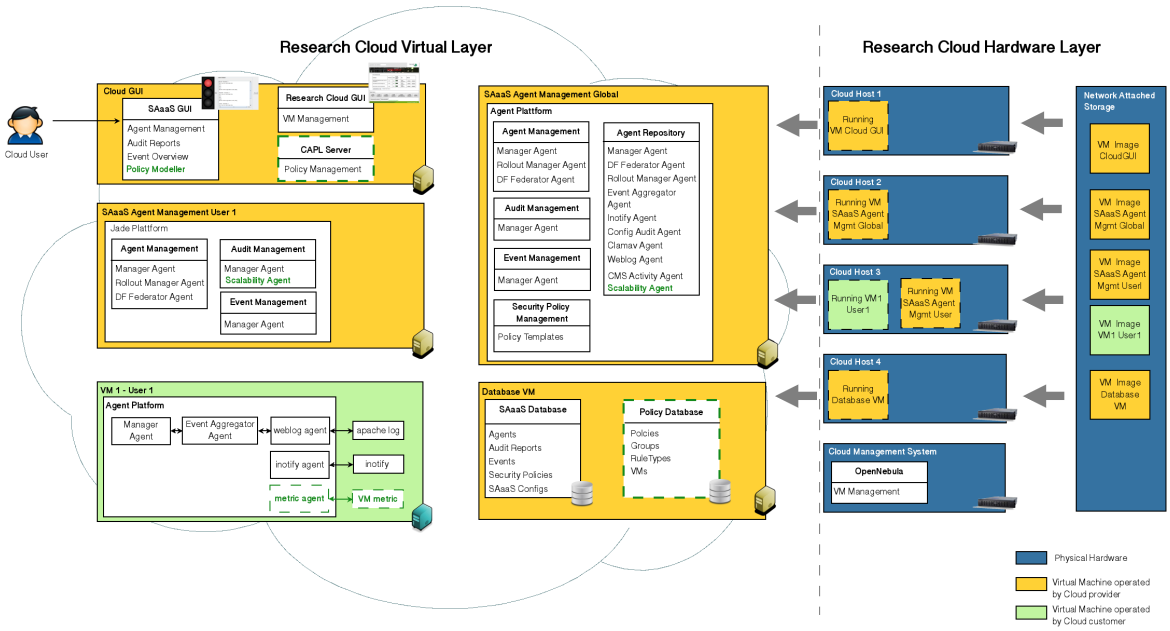


Figure 7.18: SAaaS prototype components - version 2

In the data storage backend, a CAPL database is added, storing policies, agent configurations and groups. The database got implemented, on the already existing database VM. The CAPL server is implemented as a web service, run by an Apache Tomcat application server. The cloud GUI VM was used, to host this server. The SAaaS GUI

Figure 7.19: SAaaS prototype 2 - graphical policy modeller

was extended by a graphical policy modeller depicted in Figure 7.19.

7.4.1 Server and Language Implementation

For the integration of the Cloud Audit Policy Language, existing modules of the SAaaS prototype version 1 are extended. A CAPL Server, forming the CAPL backend was developed in Java. It is based on REST [236], an architectural style used for web services. In enables data transmission and management via HTTP methods [236]. URIs are used for a distinct addressing of resources [237]. The CAPL server is provided through an Apache Tomcat [233] server. Communication is done via HTTP requests. Each request creates, modifies or deletes objects (policies or groups). Message format is implemented in XML, specified by the MIME type `application/xml`. Policies are stored in a MySQL database. The CAPL server supports the following HTTP request types:

GET - Resource Requests

Data from the server are requested. An URI defines which specific data are requested. If no parameters are specified, all existing objects are returned. The server answer

contains all objects, serialised in XML and the return value 200 - OK. A complete list of possible return values is provided in Table A.16 in Appendix A.3, Section A.3.

POST - Resource creation / Execution of operations

In case a new resource gets created, the object data get send in the data field to the server. Listing A.1 in Appendix A.3 shows a request to create an object "WWW-Cluster1". The server delivers the newly created object back and confirms the success full object creation with return value 201. The new object can now be referenced by a unique URI, such as `https://research.cloud.hs-furtwangen.de /CAPLPrototyp/res/groups/46`.

In case an operation should be executed, such as adding a `Machine` to a `Group`, a POST method with an `Action` object gets executed. The Action object contains as target the URI of the Machine, which should be added to a Group. Listing A.2 in Appendix A.3 shows the case where a machine with `id 5` gets added to group 46.

CAPL Server

The CAPL server was implemented cloud wide as a single VM running an Apache Tomcat. Its technical details are:

- **URL** - `https://research.hs-furtwangen.de/CAPLPrototyp/rest/`
- **Management** - `https://research.cloud.hs-furtwangen.de/manager`
- **Libraries** - libCAPL.jar
- **Executable** - Folder `/tomcat/work/Catalina/localhost/CAPLPrototyp`
- **Database** - onebackend

CAPL Database schema

CAPL stores all objects serialised in a MySQL database. Its schema is similar to its class diagram and depicted in Figure 7.20. `Machines` are stored in table `vms`

and MachineTemplates in table `vm_types`, due to a naming convention in the StudiCloud environment. Policies are stored in table `PolicyRule`. The table contains a column ID for either a **Machine**, a `MachineTemplate` or a `Group`. One of them needs to be set to link the policy to an instance. Each RuleType can contain multiple keys for context-based attributes. Keys are stored in table `RuleTypeKey` and can be assigned to multiple `RuleTypes`. Concrete assignment is done via a helper table `RuleType_has_RuleTypeKey` with an own unique ID for identification. This get used in table `PolicyKeyValue` for storing the raw values of a policy. Thus, it is ensured, that only attribute values, which are pre-defined by the RuleType are saved. Groups and Machines are assigned to users via its ID from table `User`. Thus, every policy inherits their attributes. Cascade deletion is executed when a group or assigned policies

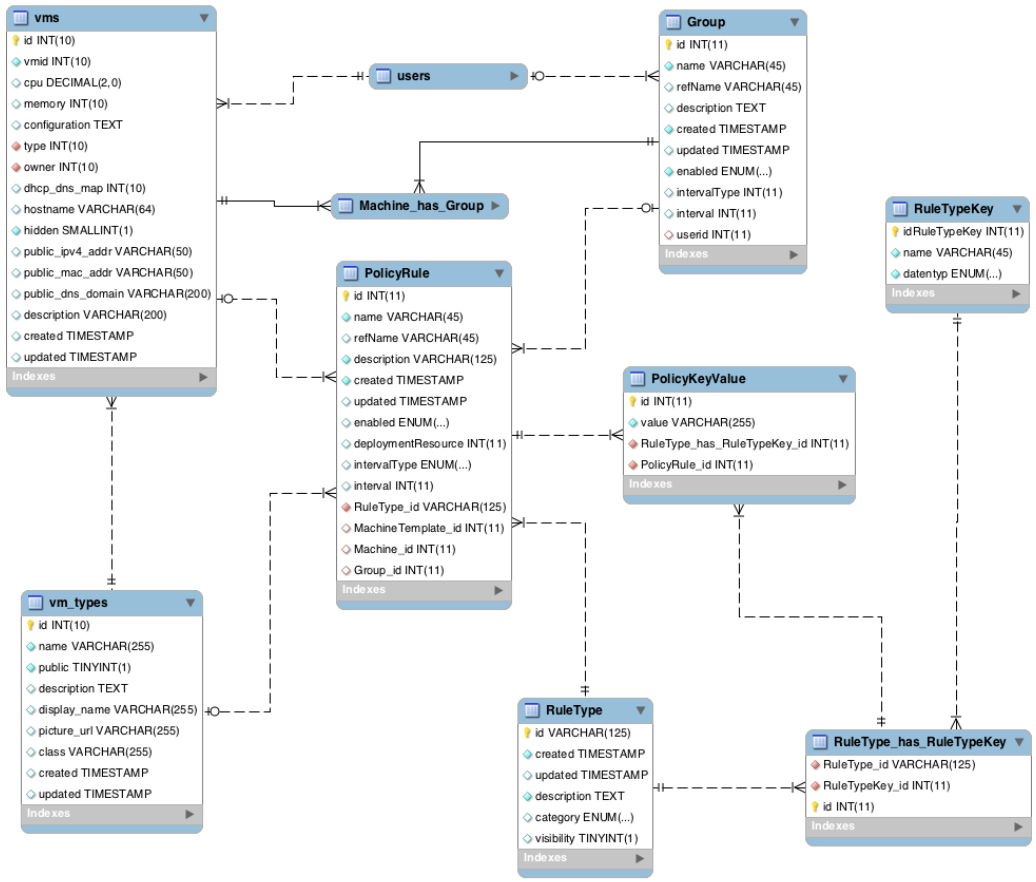
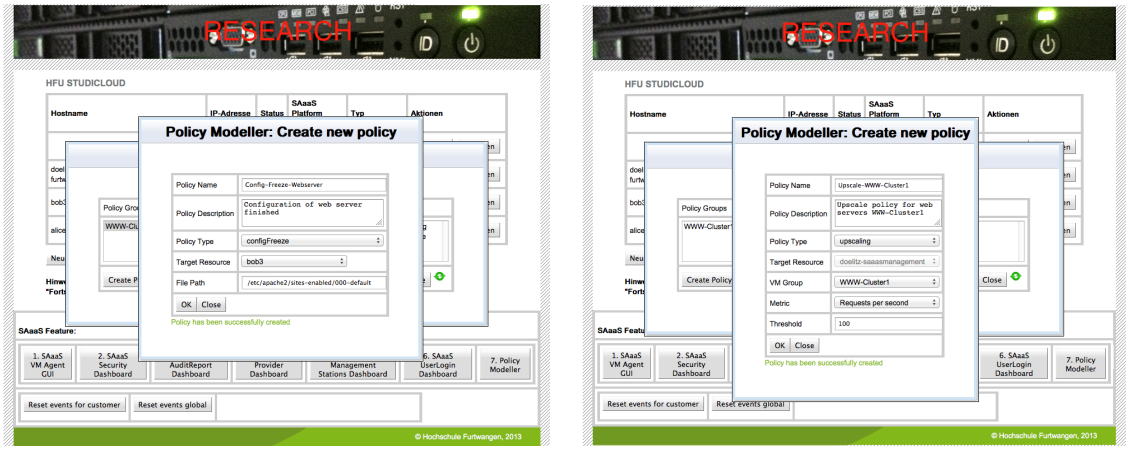


Figure 7.20: CAPL Database Design



(a) SAaaS Policy Modeller: Creation of Config Freeze policy

(b) SAaaS Policy Modeller: Creation of Upscaling policy

Figure 7.21: SAaaS Policy Modeller: Create Policy

are deleted. In case a group gets deleted, its policies get deleted as well. Machines are not altered, and exist further on. If a policy gets deleted, its assigned context-based values are deleted as well.

7.4.2 SAaaS Policy Modeller

The Policy Modeller is the graphical user frontend for cloud user (customer & provider. It is implemented in Java/GWT and depicted in Figure 7.19 and provides the following features: Create Policies, Manage Policies, Create Groups, Manage Groups and Audit Items.

Create Policies: enables the user to add new policies, see Figure 7.21. First, a policy name gets defined, and a short description of the policy can be entered. From the list "Policy Type" a policy template can be chosen. For the SAaaS prototype the following policy templates are implemented:

- ConfigFreeze: A file or directory is considered final, filesystem changes should be monitored
- Malware: The VM should not contain any malware

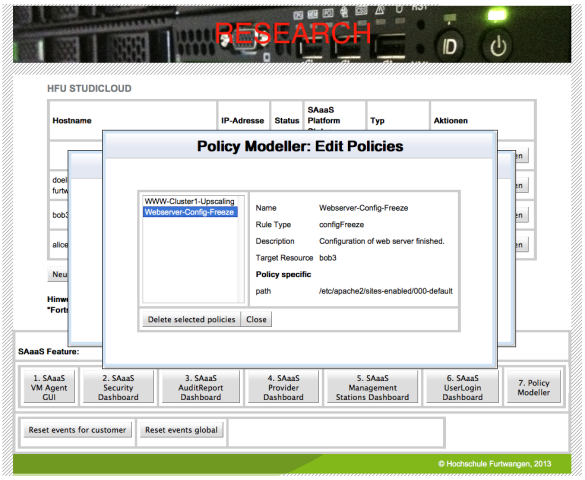


Figure 7.22: SAaaS Policy Modeller: Manage existing policies

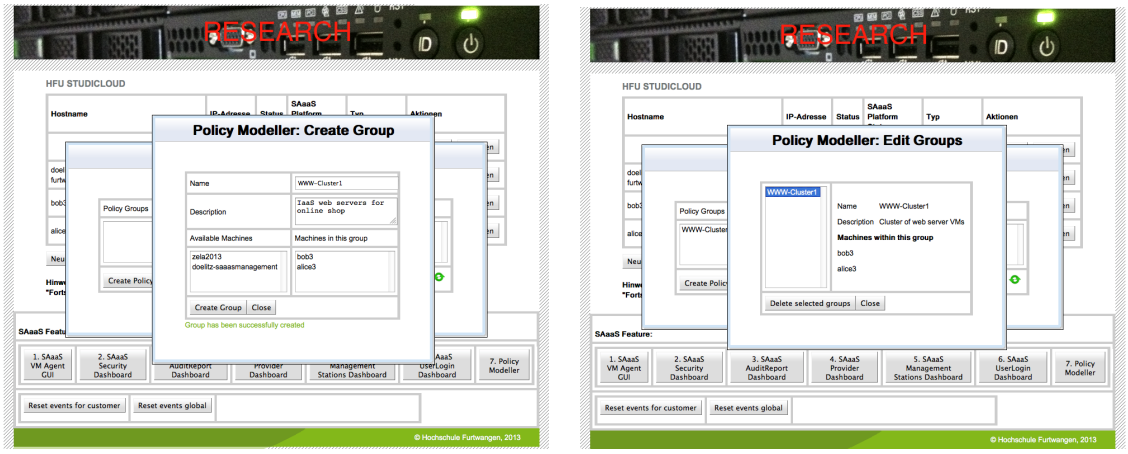
- Upscaling: Constraints for an allowed upscaling of the instance (VM or group)
- Downscaling: Constraints for an allowed downscaling (VM or group)

Depending on which policy template was selected, additional information need to be entered. In case of a Config Freeze policy (Figure 7.21(a)), the target resource (VM or group) needs to be entered. Same applies for an anti-malware policy. If an up- or downscale policy is selected (Figure 7.21(b)), a VM group has to be selected. Furthermore a metric needs to be defined, which should be checked by the SAaaS sensor agents every time an up- or downscaling event gets detected for this VM group.

Manage Policies: Allows the detailed display of existing policies, depicted in Figure 7.22. For the SAaaS prototype a *Delete policy* feature was implemented. In a productive system several modifications of existing policies will be available here.

Create Groups: Allows the creation of VM groups, depicted in Figure 7.23(a). Thus, policies can be assigned to multiple VMs logically belonging together.

Manage Groups: Allows the detailed display of existing groups, depicted in Figure 7.23(b). For the SAaaS prototype a *Delete policy* feature was implemented. In a productive system several modifications of existing groups, like adding and removing of VMs will be available here.



(a) SAaaS Policy Modeller: Creation of VM group WWW-Cluster1

(b) SAaaS Policy Modeller: Management of VM group WWW-Cluster1

Figure 7.23: SAaaS Policy Modeller: Group creation and management

Audit Items: Allows the instant auditing of selected policies. This supports the case, that a user manually wants to check the compliance of one or multiple policies or wants to execute a defined audit immediately.

Table 7.3 shows a list of agents, which were developed during this stage of the SAaaS prototype. A comprehensive list of all developed agents during this research including technical details is provided in Appendix A.2.

Agent name	Agent Type	Position	Description
Scalability Event Agent	Management Agent	SAaaS-Agent Management Platform (global & user specific)	Gets created in case a scalability request gets detected by a CMS agent. Deploys Metric Agents and collects result
Metric Agents	Metric agent	Target VM / cloud host	Checks a certain metric, e.g. CPU load or HTTP requests

Table 7.3: SAaaS agents developed during 2nd SAaaS prototype stage

7.4.3 SAaaS Demo 2 - Detection of Scalability Attacks

As for demonstration, the identified cloud specific security risk "B5. Missing monitoring of cloud scalability" (see *Chapter 3.4.3 - Specific Cloud Security Problems*) was

chosen as a demonstration scenario. Figure A.2 in Appendix A.3 presents communication between all involved components and describes it. Similar to SAaaS Demo 1 it is divided into three parts:

1. **VM preparation** - Deploy of web server VMs, and policy generation
2. **Attack on cloud management** - Malicious upscale event gets issued
3. **Evaluation & Reporting** Upscale event get evaluated and denied

Demo preparation A cloud user creates three new VMs on the web based cloud management interface, depicted in Figure 7.24. The VMs get configured as a typical Web application installation: two Web servers, one database server.

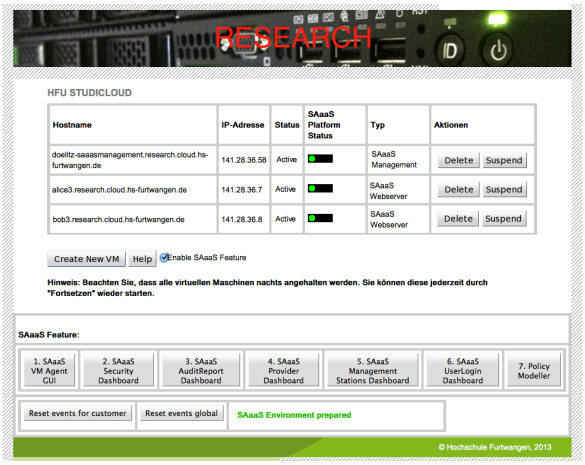


Figure 7.24: SAaaS cloud GUI: Web server VMs created

After VM configuration is finished the user enables the SAaaS features starts the security policy modeller and groups the VMs into group *"WWW-Cluster1"*. Then he activates a scalability monitoring policy with a metric of HTTP requests per second and a threshold of 100 for *WWW-Cluster1*. Furthermore, he creates a policy saying that the *"/etc/apache2"* config directory should be considered frozen and therefore be monitored for changes. As a result from enabling the SAaaS features and the policy

creation a sensor agent for filesystem monitoring gets deployed to the VMs of *"WWW-Cluster1"*. It utilizes the linux tool *inotify* [238] to watch the *"/etc/apache2"* directory.

Execution of Attack It is assumed, that an erroneous upscaling request gets issued to the cloud management system for *WWW-Cluster1*. This can be caused due to a malfunction of the CMS or by a malicious attacked. It gets intercepted by a SAaaS agent monitoring the CMS.

Mitigation by the SAaaS System The event provokes the creation of a scalability audit agent, which get deployed to the user's SAaaS Agent Management Platform. The agent creates new metric agents according to the underlying scaling policy, which get deployed to the web server VMs of *WWW-Cluster1* to check the current load of HTTP requests. They report the result back to the audit agent. The Scalability Agent evaluates the result and allows or denies dependent on the average load reported by the metric agents the requested upscale event.

Figure 7.25 shows the SAaaS security dashboard after the scalability attack demonstration. The dashboard informs that an upscaling event was detected at the cloud management system and the result of the corresponding concurrent audit. In this case scaling was denied, since no load was recognised at the web server VMs of WWW-Cluster1.

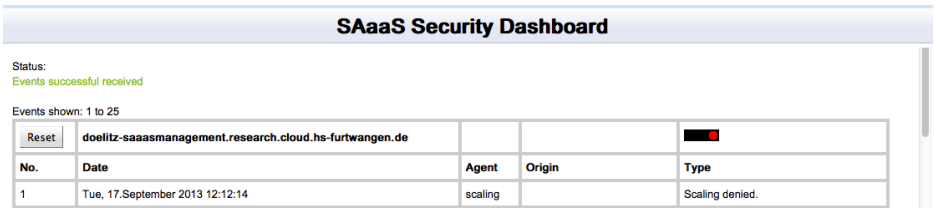


Figure 7.25: Cloud security dashboard: Mitigated scalability attack

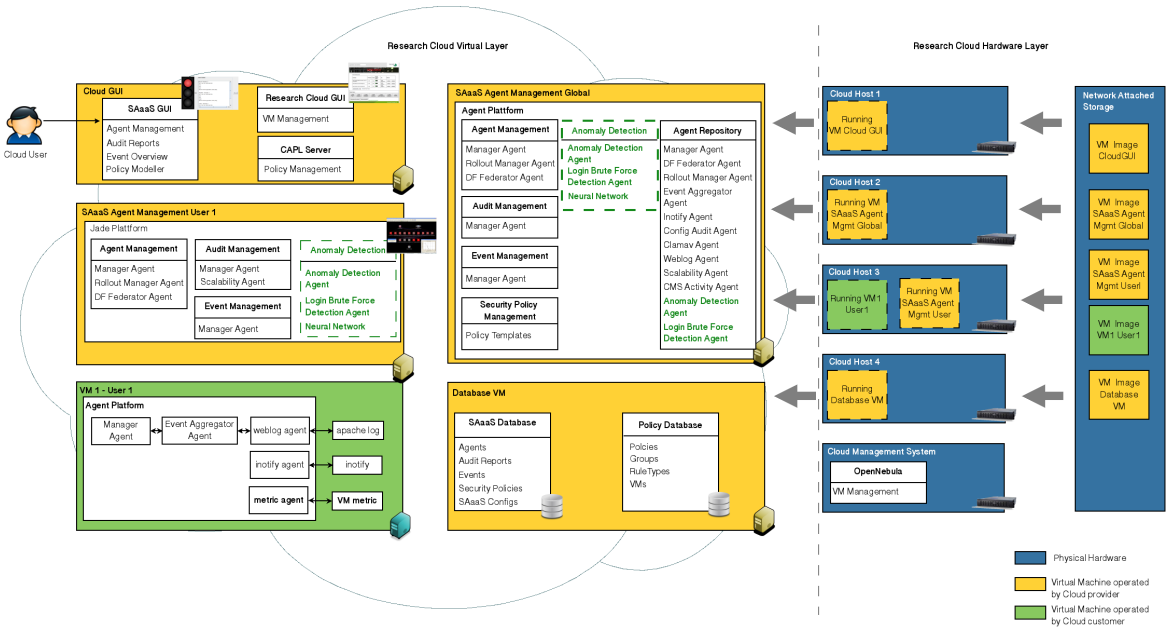


Figure 7.26: SAaaS prototype components - version 3

7.5 SAaaS Prototype 3 - Integration of Anomaly Detection

To show the feasibility of the developed anomaly detection system, presented in *Chapter 6 - Anomaly Detection*, the SAaaS prototype version 2 got extended by components of an anomaly detection system. Figure 7.26 shows the SAaaS prototype version 3 architecture, new components highlighted in green. The prototype is extended by a cloud usage data simulator, rule based policies for anomaly detection as well as a corresponding agent, and a anomaly detection system feeding a neural network to evaluate certain user behaviour.

The system is fed by a SAaaS sensor agent, monitoring the cloud management system. Whenever a VM-CREATE or VM-DELTE command is monitored for a certain user, this is forwarded to the anomaly detection system. The user's behaviour profile is loaded and the monitored event is tested against its behaviour profile by the neural

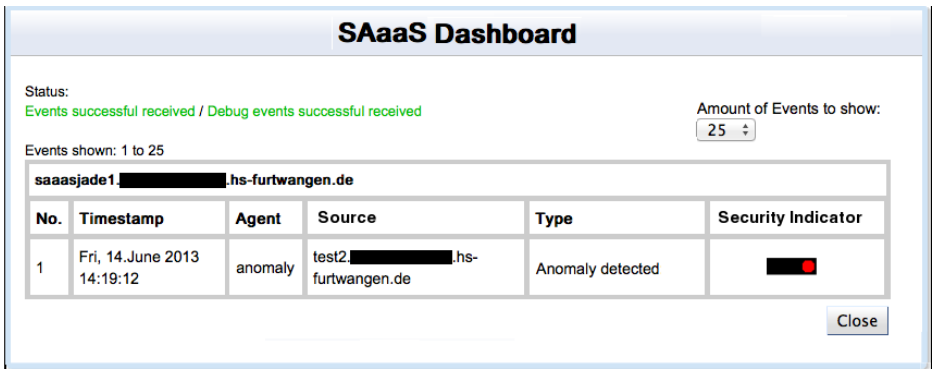


Figure 7.27: Anomaly warning in SAaaS Dashboard

net. In case an anomaly was detected, this gets reported back to the SAaaS agent, and a warning is displayed in the SAaaS Security Dashboard in the web based cloud management GUI of the user. Figure 7.27 shows a screen shot of such a warning.

7.5.1 Agent Development for Anomaly Detection

Similar to *Chapter 6 - Anomaly Detection*, integration of the anomaly detection functionality into the SAaaS prototype is implemented as two different agents:

- Rule based detection - Login Bruteforce Detection Agent
- Behaviour based detection - Anomaly Detection Agent

Rule based detection - Login Bruteforce Detection Agent Technical functionality of defining rules of an expected, normal behaviour is already provided in the SAaaS prototype through the cloud Audit policy language. To demonstrate a rule based detection scenario, a Login Bruteforce Detection agent is implemented. It gets deployed to each VM of a cloud users, monitoring SSH logins and login attempts. To avoid double description, its functionality is described directly in *Subsection 7.5.2 - SAaaS Demo 3 - Login Bruteforce Detection*.

Behaviour based detection - Anomaly Detection Agent To show the functionality of a behaviour based anomaly detection system an Anomaly Detection Agent is

implemented. It gets deployed to the user specific SAaaS Agent Platform. It gets information about VM action from the CMS Activity Agent about newly created VMs, starting and stopping of existing VMs, or deletion of a VM. This information gets forwarded by the Anomaly Detection Agent to a neural network. For the SAaaS prototype version 3, this is implemented on a separate Windows VM running the neural network editor Membrain. For a productive system, this would be implemented directly at the SAaaS Agent Management Platform, thus it is placed there in Figure 7.26. User concentric anomaly detection is performed on the user specific SAaaS Agent Management Platform, whereas cloud wide anomaly detection is done by the provider on the "SAaaS Agent Management Platform Global". Each network is trained with historical user data and evaluates the newly received event if it indicates an anomaly or not. This information gets passed back to the Anomaly Detection Agent.
Table 7.4 shows a list of agents, which were developed during this stage of the SAaaS prototype. A comprehensive list of all developed agents during this research including technical details is provided in Appendix A.2.

Agent name	**Agent Type**	**Position**	**Description**
Login Bruteforce Detection Agent	Sensor agent	Target VM / cloud host	Monitors Login Attempts
Anomaly Detection Agent	Sensor Agent	SAaaS-Agent Management Platform (global & user specific)	Feeds behaviour based anomaly detection system

Table 7.4: SAaaS agents developed during 3rd SAaaS prototype stage

7.5.2 SAaaS Demo 3a - Login Bruteforce Detection

To demonstrate rule based detection of cloud wide anomalies, the attack scenario of a distributed login attack, elaborated in *Chapter 6.3.2 - Cloud Wide Anomaly Detection* was chosen. As a reminder, Figure 6.2 (Chapter 6.3.2) depicts the attack: An IP range of cloud resources gets brute forced by an attacker with one specific

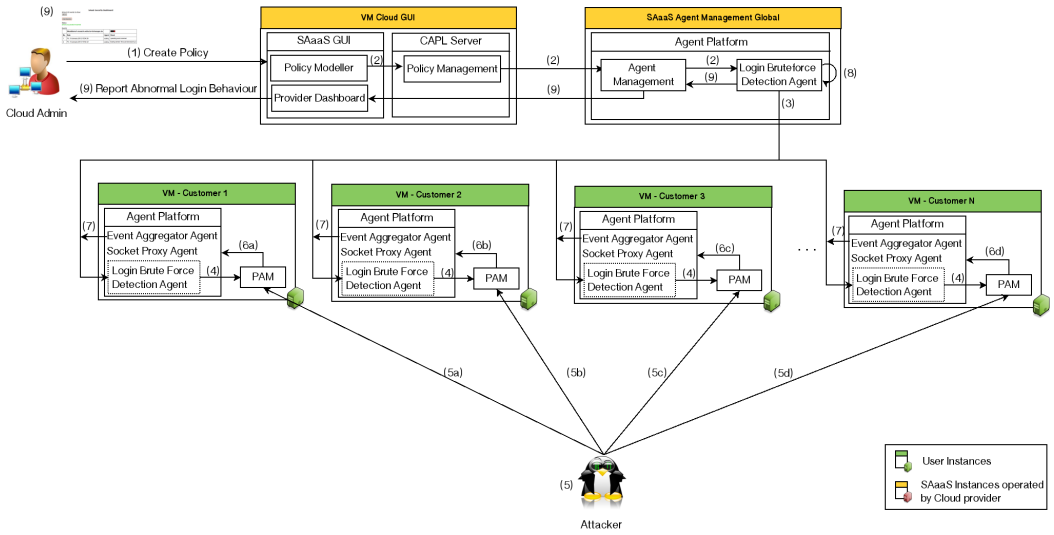


Figure 7.28: SAaaS prototype demo 3a: Detected abnormal login

username+password combination. Weakly secured VMs using default user credentials such as *username: demo, password: demo* are vulnerable. The demonstration, is divided into four parts:

- **Preparation** - cloud provider defines cloud wide security policy
- **Attack** - Attacker performs distributed bruteforce login attack
- **Detection & Evaluation** - SAaaS agents detect attack
- **Reporting** - Detected abnormal behaviour gets reported

Figure 7.28 visualises the sequential process of SAaaS Demo 3a.

Preparation - It is assumed, that a cloud provider is running an IaaS Cloud. Several customers are running VMs in this Cloud. One of them is weakly configured with a possible SSH account (username: root, password root). To secure the infrastructure, the cloud provider defines the following cloud wide security policy ①: "If a successful SSH login of a specific SRC system is preceded by certain threshold of unsuccessful login attempts at other VMs in the cloud infrastructure, this is considered an anomaly and an alarm should be raised". For this demo, a threshold of five login attempts was

set. The threshold gets calculated as follows: If an unsuccessful login attempt, prior to the successful login was at a VM from the same cloud customer, a value of one gets added to the threshold counter. If the prior login attempt was recognised at a VM of a different customer, the threshold gets increased by three. This calculation respects the possibility of false positive login attempts, where a user might mistypes a password, whereas it is considered very unlikely, that a user might try to login at an instance of a different customer. The definition of the providers cloud wide policy results in the following automated actions:

- The policy gets executed by the CAPL server. A Login Bruteforce Detection Agent gets started at the Global SAaaS Management Platform (2a).
- It forks itself to all SAaaS-enabled VMs running in the cloud (3). At each VM it updates (4) the local Pluggable Authentication Modules (PAM) to send a notification to the local Socket Proxy Agent about SSH login attempts. Afterwards, it deletes itself from the VMs. Each login attempt at a VM gets now reported through the local Socket Proxy Agent and the local Event Aggregator Agent to the Login Bruteforce Detection Agent.

Attack Now, a distributed login bruteforce attack gets demonstrated (5). Therefore, a Linux bash script was developed, trying a SSH login with a fixed username and password combination at a list of cloud instances (5a) - (5d). Through the order of VM names in the list it is made sure, that the weakly configured VM is accessed last.

Detection & Evaluation The login attempts (unsuccessful and successful) get recognised by the local PAM modules ((6a) - (6d)) and forwarded (7a) - (7d) to the Login Bruteforce Detection Agent. For each login src (attacker), a login counter gets created, which increases at a failed login. If a successful login is recognised, the number of unsuccessful logins at different VMs gets evaluated. If the counter exceeds the thresh-

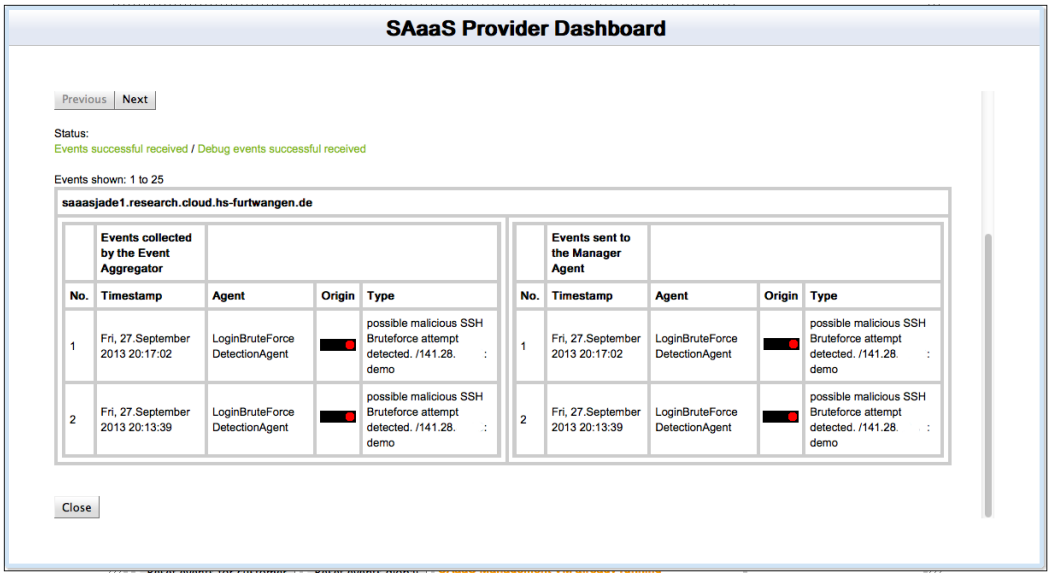


Figure 7.29: SAaaS prototype demo 3a: Detected abnormal login

old defined by the cloud provider's policy, this gets reported to the SAaaS Security Dashboard. Since three unsuccessful login attempts at three different customer VMs exist for the same SRC in this demonstration, the login counter for this attacking host is nine, exceeding the limit of five. Thus, an alarm gets issued by showing a message in the SAaaS Provider Dashboard.

Reporting The cloud provider gets notified about the abnormal login behaviour through the SAaaS Security Dashboard. Figure 7.29 shows the event in the SAaaS Provider Dashboard. For the SAaaS demo, no further actions are defined, but in a real system further incident prevention methods, such as blacklisting a SRC system are imaginable.

7.5.3 SAaaS Demo 3b - Abnormal VM Creation

To demonstrate Behavioural based detection of cloud anomalies, the user concentric behaviour analysis scenario, presented in *Chapter 6.4.4 - User Concentric Anomaly Detection With Neural Networks* is implemented into the SAaaS prototype, forming

prototype version 3. Therefore, for each user, a behaviour profile of VM CREATE and VM STOP events is built up. Furthermore, an anomaly detection module, operating a neural network is added to the user specific SAaaS Agent Management Platform (see Figure 7.26). Figure 7.30 visualises the developed SAaaS demo 3b.

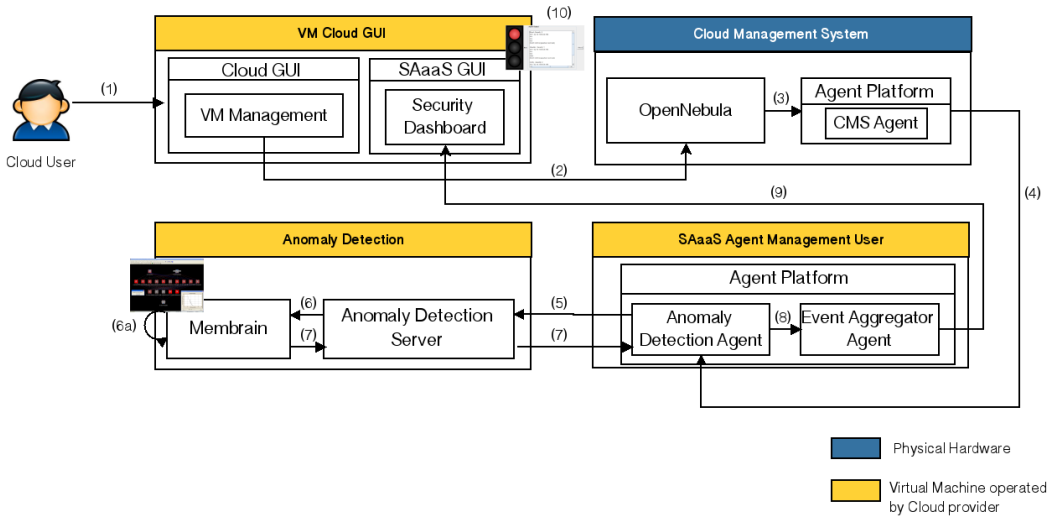


Figure 7.30: SAaaS prototype demo 3b: Anomaly detection for VM life cycle behaviour

No pre-assumption is taken for this demonstration. First, a cloud user creates a new VM ①. The request gets issued to the Cloud Management System ②. An Anomaly Detection agent, located at the user specific SAaaS Agent Platform gets informed by the CMS agent ④ whenever a user creates or stops a VM. This event gets forwarded to the Anomaly Detection Server ⑤, the neural network gets started, learning the user's prior VM usage behaviour ⑥a from its prior recorded behaviour profile. When the learning phase is completed, the new event gets assessed by the network to classify, if this event fits to the user's learned "normal" behaviour. In case it fits, no further actions are taken. If it does not fit, this is considered an anomaly, and an appropriate alert action is executed ⑦. Therefore, the existing SAaaS communication path is executed: The event gets forwarded from the sensor agent (Anomaly Detection Agent) to the Event Aggregator Agent, which issues a warning ⑨. For the demo, an alert is

shown in the SAaaS Security Dashboard (10), depicted in Figure 7.31.

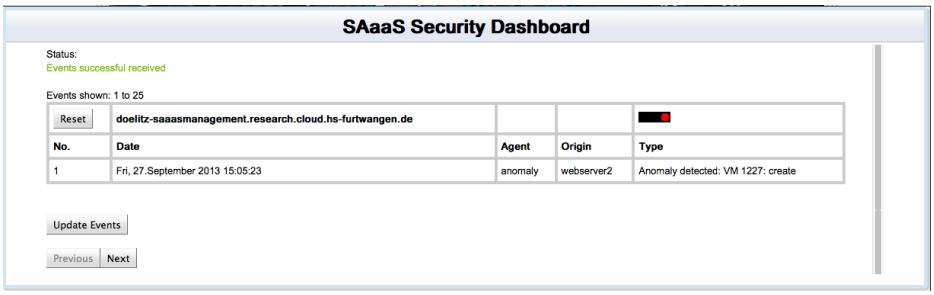


Figure 7.31: SAaaS prototype demo 3b: Abnormal VM creation detected

For the SAaaS prototype version 3, the 4-40-20-10-5-1 neural network for scenario UserAnomaly03 (see Chapter 6.4.4) was used, implemented using the neural network editor Membrain, running on a separate Windows VM. Membrain was chosen due to its scriptability. A python network server was developed, opening a TCP socket, waiting for an event from the Anomaly Detection agent about a VM CREATE or VM STOP message. Figure 7.32 shows some debug output of the python server.

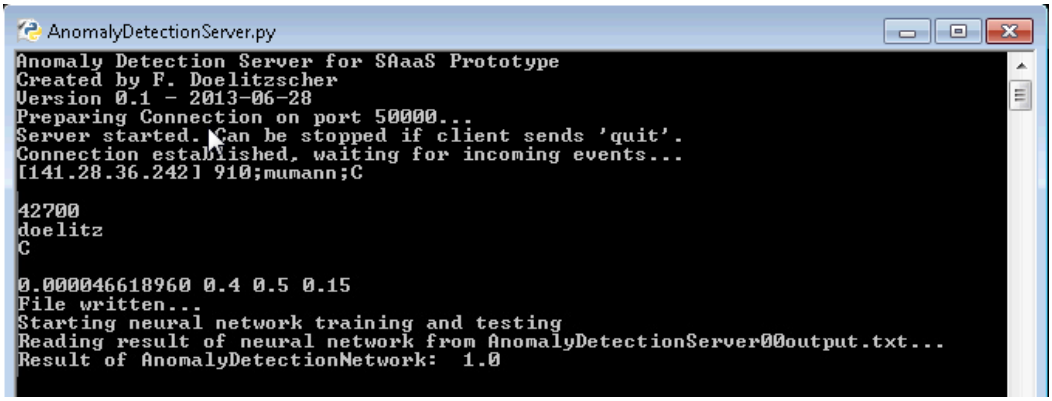


Figure 7.32: SAaaS prototype demo 3b: Anomaly server output

A VM CREATE event (C) is sent from the Anomaly Detection Agent to the server, together with the user id (doelitz) and a time stamp in form of seconds of that day (42700), which corresponds to 11:51am. This information gets forwarded to a membrain script, which:

1. normalises the data as test data

2. loads and trains the network with historical usage data of user “doelitz”

3. evaluates the test data against the learned behaviour profile

4. reports back a anomaly result (0 or 1)

Figure 7.33 shows the neural network, during the training phase after the received VM CREATE event of user doelitz. As depicted in the lower part of Figure 7.32, this event is considered an anomaly (result evaluation 1.0). This is, because for the demonstration, a cloud usage history for user doelitz was prepared, stating, that the user normally only creates VMs during 2am - 4am in the morning. Thus, a VM creation at 11:51am does not fit to the learned behaviour.

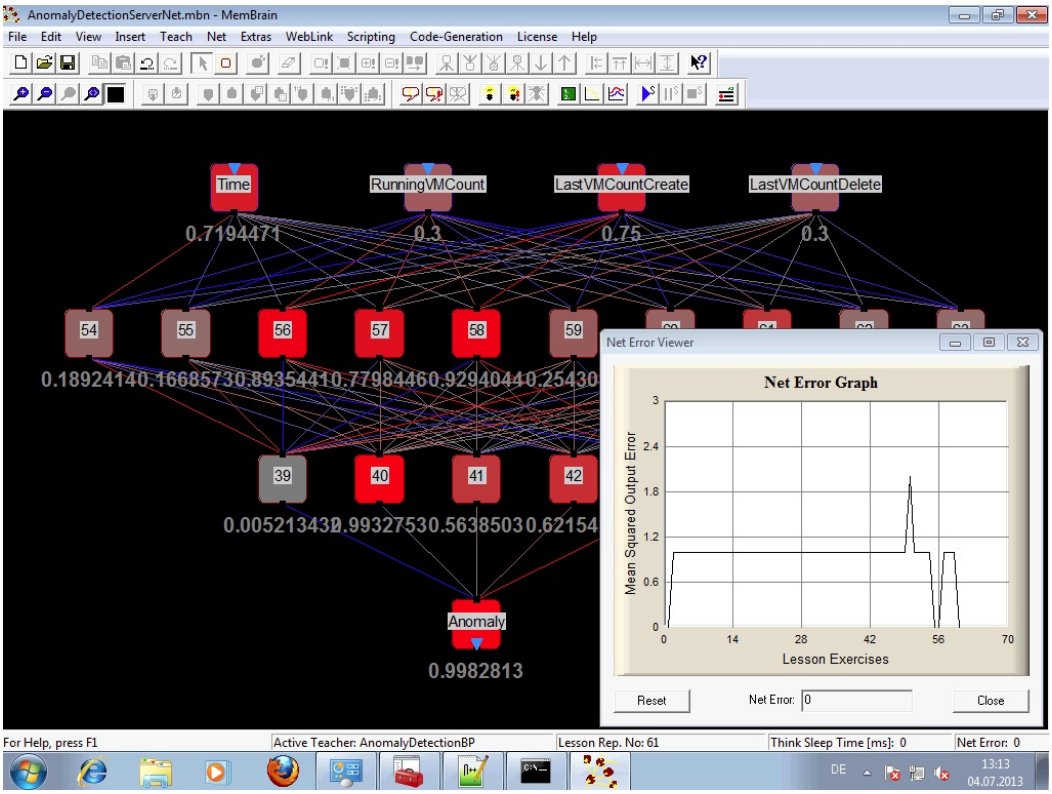


Figure 7.33: SAaaS Prototype 3 - Training of user behaviour

7.6 Summary

This Chapter presented a three-stage prototype of the Security Audit as a Service system. The technical environment CloudIA was introduced and the integration of the JADE agent framework presented. SAaaS demo 1 showed how automatic concurrent audits react on a change within a Cloud infrastructure. As a second development stage, a prototype of the Cloud Audit Policy Language was presented, which got included into the SAaaS prototype. SAaaS demo 2 presented the graphical policy modeller and elaborated how the SAaaS system mitigates scalability attacks. As a third extension stage, the presented anomaly detection system was implemented into the prototype. SAaaS demo 3 showed how rule based and behavioural based anomaly detection is performed. The prototype validates the feasibility of the presented research.

8

Evaluation of the SAaaS architecture

"A delusion is something that people believe in despite a total lack of evidence."

(Richard Dawkins, ethologist, evolutionary biologist)

This chapter evaluates the developed SAaaS architecture both theoretically and practically to prove, that the work on Security Audit Compliance For Cloud Computing is a valid solution to mitigate the identified cloud security issues.

8.1 Introduction

This chapter evaluates the presented research. First, it is elaborated how the developed SAaaS architecture improves cloud audits and how it brings more transparency into IaaS cloud infrastructures. Afterwards it is evaluated, which cloud specific security issues are addressed and mitigated by the research.

8.2 How SAaaS improves Cloud Audits

Using an agent based audit system in cloud environments is of advantage because of the adaptability to many different instances hosting many different services. Small agents, programmed for one single task are using very few resources and when finished with the task the agent can be deleted. Specific sensor agents can be utilised by multiple other agents to become a clearer picture about the current security state of their surroundings. Audit agents are moved to a VM to e.g. check configuration compliance to security policies.These audits can be made dynamically on a specific VM or a given range of VMs to check the security status. They are done by request from a user or as an automatic reaction to an event triggered by an agent on a monitored VM. If such a security problem is detected at one VM an audit on similar types of VMs can be done. The following shows, how SAaaS improves security audits of cloud infrastructures.

8.2.1 Cloud Monitoring and Audit

Cloud customers so far have just limited possibilities to monitor their cloud instances. This leads to a problem of lack of trust in cloud computing technology and provider. In a SAaaS-enabled cloud infrastructure, user VMs are equipped with agents. Users define security policies, describing which VM components are to be monitored, which

behaviour of this VM is considered "normal" and how to alert in case of system security suspicion. The status gets conditioned in a user-friendly format accessible easily through a web portal – the SAaaS Security Dashboard. Continuous monitoring creates transparency about the security status of a user's cloud instances, hence increasing the user's trust into the cloud environment. Furthermore, with SAaaS agents monitoring at key points in the infrastructure of a cloud, customers can be warned if a security problem occurred in their cloud instances environment, for instance, in a VM which is running on the same cloud host. The user can model with the Cloud Audit Policy Language in a fine-grained manner, if they want to take a certain risk or if further actions are required to protect their instance. This could result, for example, in the migration of a VM to a different cloud host or a shutdown of the VM and the start of a twin VM at a different cloud data centre.

8.2.2 Message Reduction and Aggregation of Raw Data at Side to Audit Data by Business Process Awareness

A traditional IDS would produce many messages when monitoring a cloud environment due to a lack of flexibility regarding frequent infrastructure changes. With SAaaS, in case a monitoring event is produced, it first is processed by the agent, which is initiating the event. Afterwards, this agent informs all other agents (mainly the Event Aggregator Agent), which are also involved in the current business case. This is important to reduce the overall messages sent to the cloud event processing system especially in large cloud computing environments. Imagine an expected high load on the load balancer can result in a high number of events produced by the load balancer's agent. Since the events are expected, they again result in a high load on the web server and the database whose corresponding agents could produce again a high number of events. By informing the business flow participating agents (Weblog Agent, Database

agent) with an abstract message, false- positive event messages will be prevented. For cloud audit, this can also be a possibility to prevent data storms by using special audit aware agents.

8.2.3 Automated Auditing of VM Images

To evaluate the automatic audit system the following scenario is considered: A cloud customer chooses an online shop virtual appliance (containing a web server and a database) out of a cloud's store. Before transferring actual data to the image the following security policies need to be evaluated:

- P1: The image must not contain any malware
- P2: The image must not run any other software than the web server and the database
- P3: The web server and database connection must be configured properly

To evaluate the SAaaS approach of concurrent audits it is compared to the alternative of a manual audit approach.

Customer uses Automatic Audit System Approach

When using the automatic audit system, the appliance user or appliance creator first describes his security policies. Since malware checks are usually a default policy, provided by the cloud provider, there is no need to model those. Additionally, to simplify black box scans of the proper interaction between web server and database it is imaginable to deposit a predefined default start page, which could be browsed. The automatic audit system will parse the security policies and identifies the necessary audit cases, which are fetched from the database. The audit cases get sorted, dependent on how the checks can be executed. There are two kinds of security audit modes. *Offline VM audits* mount the VM's image and perform audit tasks on it,

whereas *online VM audits* launch the VM in a quarantine environment of the cloud. There, audit tasks, which can be only performed on the running VM are executed, such as an analysis of open ports. Again, the results of the single audit cases will be submitted to the parser and saved as mini reports in the audit system's database.
At last, the report generator conditions the results of all mini reports. The cloud management system is informed if the overall audit result is *"passed"* or *"failed"*. If the status is passed, the image can be added to the store, otherwise the release will be denied. Nevertheless of the result, the complete audit report will be sent to the appliance creator, to inform about necessary problems to be fixed.

Manual Approach

In contrast to the automatic audit system an offline audit of the appliance's image is not immediately possible. This is due to the fact, that the appliance user does not have direct access to images stored in the appliance store. The only two approaches possible are downloading the appliance's image, which enables offline auditing or to limit the audit process to online auditing. This is done by an administrator, who must have sufficient expertise in virtualization technologies, auditing methods and must be an audit tool expert. Additionally, for the sake of reproducibility and documentation, the appliance user has to follow a very well defined auditing process (assuming such a process exists). Downloading VM images and evaluating them offline imposes a significant network overhead on the appliance user as well as the cloud provider. Manual online auditing can be performed, when a virtual appliance image is already started. The appliance user has to log in to the appliance in order to execute auditing tools and scripts. Additionally, the virtual appliance also has to be checked externally to determine which services are activated, for example by port scanning. Performing port scans on virtual machines executed in the cloud may trigger the cloud provider's intrusion detection systems or may even be prohibited entirely by the cloud provider's

terms of use. This demonstrates the overall complexity of a manual virtual appliance auditing process. The automatic audit system delivers the following advantages for appliance creator/user and cloud provider: Improved security when using 3rd party virtual appliances (appliance user), well documented and formalised audit process (all), customizable, machine-readable audit policies (all) and additional revenue by offering audits as a service (Cloud provider).

8.2.4 The Cloud Audit Test Criteria Catalogue

One major result of the research on cloud security issues and cloud audits is the Cloud Audit Test Criteria catalogue, presented in *Chapter 3.5.4 - Cloud Audit Test Criteria Catalogue.* This catalogue presents over 140 test criteria for IaaS cloud environments. It provides cloud user and cloud provider with a checklist to evaluate their cloud infrastructure against cloud security issues. Despite existing cloud security guidelines, it provides a specific question set on technical and organisational parameters how a cloud infrastructure is operated. To the best knowledge of the author, such a list does not exist yet. In 2013, the catalogue was downloaded over 32 times. It is used by the commercial security service provider SCHUTZWERK GmbH [138] to support IT security audits of cloud infrastructures. Furthermore, the IT Security manager of the data centre of the federal state of Rhineland-Palatinate (Germany) officially requested usage rights on the catalogue and the corresponding audit criteria checklist. The data centre hosts an ISO 27001 [83] certified cloud infrastructure for the public state administration and the state police. It plans to use the catalogue and develop it further.

8.3 How SAaaS Addresses Cloud Security Issues

To show the feasibility of the developed agent based Security Audit as a Service architecture, first a comparison to the identified cloud security risks (Cloud specific and amplified), introduced earlier in Chapter 3.4 is presented.

8.3.1 Mitigation of Cloud Resources Abuse

Cloud computing advantages are also used by hackers, enabling them to have a big amount of computing power for a relatively decent price, startable in no time. Cloud infrastructure gets used to crack WPA, and PGP keys as well as to host malware, trojans, software exploits used by phishing attacks or to build botnets like the Zeus botnet [79]. The problem of malicious insiders also exists in classical IT-Outsourcing but gets amplified in cloud computing through the lack of transparency into provider process and procedure. This issue affects authorisation, integrity, non-repudiation and privacy. Strong monitoring of user activities on all cloud infrastructure components is necessary to increase transparency. Thus, the proposed behaviour analysis of cloud usage, discussed in *Chapter 6.4 - Behavioural Based Anomaly Detection* addresses the cloud security issue B2 - Abuse and nefarious use of cloud resources (*Chapter 3.4.3*). One reason, of using a cloud infrastructure is to benefit from its scalability attributes. In this context it is most often used to deal with usage peeks, for example if a new version of a software gets released and huge download requests are expected. Characteristic to peeks is that they are mostly foreseeable and limited to a certain time frame. Therefore, cloud users design their cloud application to start new instances if a certain threshold is reached to provide service availability. This results in two challenges for cloud security:

IaaS upscaling - business driven: Since a user's infrastructure can change rapidly (grow, shrink) in case of a peek scenario the incident detection system needs to be

aware of the peek situation and the defined scalability thresholds. Thus, false positive incident alarms can be avoided if service requests due to newly created VM instances get detected. The monitoring system is aware of the changing infrastructure.

IaaS upscaling - attacker driven Most of the time, scalability thresholds like "maximum number new VMs to be created" get defined once. Mostly during the virtual infrastructure setup time, taking experience of utilisation or expected peak utilisation. If the peek was managed well by the thresholds they just stay, like defined, although they might be not needed anymore (e.g., until the next software release). This allows a new cloud specific attack: Financial damage due to nefarious abuse of cloud resources. Attackers can cause the creation of new cloud instances up to the scalability threshold by creating a huge number of allowed requests, which do not result in any successful business case e.g., distribution of malicious software. Cloud monitoring needs to be aware of business processes to detect an event of possible misuse of cloud scalability. By enhancing agents with business based usage of cloud resources, cloud security problems *B2 - Abuse and nefarious use of cloud resources* and *B5 - Missing security monitoring of cloud scalability* are addressed by the Security Audit as a Service architecture.

8.3.2 Mitigation of Shared Technology Issues

In cloud computing, isolation in depth is not easily achievable due to usage of rather complex virtualization technology like VMware, Xen or KVM. Persistent storage is shared between customers as well. Cloud provider advertise reliability measures to pretend data loss like replicating data up to six times [239]. In contrast customers have no possibility to prove if all these copies get securely erased in case they quit with the provider and this storage gets newly assigned to a different customer. While the presented SAaaS architecture does not directly increase isolation in depth it adds to

the detection of security breaches by helping to contain the damage by the presented actions. Security breaches on cloud hosts or VMs can be detected by sensor agents. Customer can define audits, which get automatically executed. Besides the presented alerting, the SAaaS system also enables any kind of automatic execution of tasks, in case of a detected security breach. An example could be a preliminary shutdown or migration of virtual instances to a different Cloud provider to contain service integrity. This addresses cloud security issue *A4 - Shared technology issues*.

8.3.3 Better Cloud Monitoring

In a SAaaS enabled IaaS cloud infrastructure, user VMs are equipped with SAaaS agents. The user uses CAPL to describe, which VM components are to be monitored, which behaviour of this VM is considered "normal" and how to alert in case of system security suspicion e.g., open network connection without a preceding legitimate request. Concurrent audits are evaluating the infrastructure's security state. Results get conditioned in a user friendly format in a web portal - the SAaaS Security Dashboard. Continuous monitoring creates transparency about the security status of a user's cloud instances hence increasing the user's trust into the cloud environment. This addresses the cloud security problem *A2 - Missing transparency of applied security measures*.

Security incidents in cloud environments occur and (normally) get fixed by the cloud provider. But, to the best knowledge of the author no cloud provider so far provides a system, which informs user promptly if the cloud infrastructure gets attacked, enabling them to evaluate the risk of keeping their cloud services productive during the attack. Thereby the customer must not necessarily be a victim of the attack, but still might be informed to decide about the continuity of his running cloud service. Furthermore no cloud provider so far shares information about possible security issues caused by

software running directly on cloud host machines. In an event of a possible 0-day exploit in software running on cloud hosts (e.g., hypervisor, OS kernel) cloud customer blindly depend on a working patch management of the cloud provider. The presented automatic auditing system of cloud instances addresses this problem, listed as *A2 - Missing transparency of applied security measures*.

Fine-grained modelled CAPL policies combined with SAaaS host agents could warn a user if a cloud provider neglects its duty of patch management to software running on cloud hosts, for example, hypervisor software. This addresses the cloud security problems *A1 - Misuse of administrator rights*, *A2 - Missing transparency of applied security measure*, and *B1 - Intransparent data location*. It can also mitigate problem *A4 - Shared technology issues* and *A5 - Data life cycle in case of provider switch or termination*.

With the presented SAaaS system, the security state of the entire cloud environment, especially the cloud management system can be monitored. Of interest are customer data and data path, administrative actions concerning customer's instances such as, patch management, incident response time, backup restore time. Continuous monitoring, combined with concurrent auditing as well as the standardized reporting of the SAaaS agents help customers to ensure the compliance of IT security best practices and help them to fulfil their responsibility to data protection laws, such as ISO 27001 [83] or BSI IT Baseline Protection [101]. It also helps cloud provider to prove compliance to IT security best practices and laws to customers and third-party IT security service providers. This is necessary for IT forensics if a security incident need to be tracked over time over multiple cloud instances or cloud hosts. This could lead to a possible cloud security certification. Thus, cloud security problem *B3 - Missing monitoring in cloud infrastructure* is addressed by the SAaaS architecture.

Table 8.1 summarizes the presented cloud security issues addressed by the presented research. Column "reference" states cloud guidelines or best practices recommendation

Cloud Security Issue	Cloud Security Threats	Proposed Mitigation	
		Cloud Audit Test Criteria	SAaaS Architecture
A1 - Misuse of administrator rights	Malicious insider[87, p16], Security Guidance for Critical Areas of Focus in Cloud Computing [86], Cloud provider malicious insider - abuse of high privilege roles[74, p36]	Access of cloud provider personnel - A37	CAPL, inotify Agent, Audit Agent, Cloud Host Agent
A2 - Missing transparency of applied security measures	Information Management and Data Security [86, p50], cloud Computing Security Risk Assessment[74, p28]	IaaS specific test criteria A37 - A40	Sensor Agents, Audit Agents, Security Dashboard
A4 - Shared technology issues	Shared technology vulnerabilities[87, p21], Security Guidance for Critical Areas of Focus in Cloud Computing [86], Isolation Failure[74, p35]	Distributed over all criteria categories	Sensor agents, Audit Agents, CAPL, Anomaly Detection System
B1 - Intransparent data location	Legal Issues: Contracts and electronic discovery [86] Risk from changes in jurisdiction[74, p45]	cloud backend storage system A20 - A26	inotify Agent, Sensor Agents (VMs, Host, Network Devices), CAPL
B2 - Abuse and nefarious use of cloud resource	Abuse of cloud services [87],	Existence of a customer overspanning misuse detection system - A36	Rule based & Behavioural based anomaly detection system
B3 - Missing monitoring in cloud infrastructure	Information Management and Data Security [86], Loss of governance[28][74], Undertaking malicious probes or scans [74, 42]	Distributed over all criteria categories	Sensor Agents, cloud Security Dashboard, Audit Agents, anomaly detection system
B5 - Missing monitoring of cloud scalability	Resource exhaustion[p33] [74]	cloud Management System A9 - A15	CMS Agent, Scalability Agent, Metric Agents

Table 8.1: Cloud Security Issues addressed by SAaaS

documents about the identified risk. Column proposes mitigation shows first, how the issue is addressed by the Cloud Audit Test Criteria catalogue and secondly which component of the SAaaS architecture addresses this issue.

8.3.4 Detection of Account Misuse

Account misuse is a well known problem from the area of web application and thus also affects the cloud's management web interface. An attacker can gain access to a victim's account by different means like hacking weak user credentials or exploiting

security flaws present in the application [206]. In cloud computing however access to an account does not only grant access to the victim's data but also to the data hosted on services which runs under the victim's account and thereby potentially to data of many service users. Additionally the attacker could spawn multiple new cloud instances, using them for his own malicious intents and thus financially damaging the victim as he has to pay for the instances. By bringing down the victim's services an attacker could potentially cause even more damage. Monitoring the user's behaviour for anomalies can help detecting account misuse. For example heavy activity on a rather silent account would cause suspicion, especially if the login sourced from another country or continent. Additionally to the VM count and geographical login source, the weekday, daytime as well as usage of available VM types gets monitored.

8.3.5 Detection of Distributed Login Bruteforce Attack

A cloud infrastructure can run a huge number of systems, maintained by different cloud users resulting in a heterogeneous level of security configuration. This is very attractive for misuse. A traditional SSH bruteforce attack tried numerous combinations of `username:password` couples on one target IP. Detection is fairly simple since this behaviour results in massive "login denied" messages in the SSH server's logfile. Therefore modern SSH bruteforce attacks are carried out by compromised computers of a botnet (bots) which just try one or two username:password combinations at a specific target and then move on to a next one, depicted in Figure 6.2. Especially huge cloud infrastructures are very attractive targets to this attack, since all systems normally are within one IP range. If one vulnerable system was found (malicious ssh login was successful) this can compromise the whole cloud security state. This attack attempt mostly stays undetected in traditional host based monitoring systems. By monitoring the login attempts, cloud wide over multiple cloud customer instances, a

successful ssh login with preceding unsuccessful ssh login attempts at different cloud instances can be defined as anomalous behaviour and thus detected.

8.3.6 Detection of VM Breakout

Several hypervisor vulnerabilities have been found in the past that allow an attacker to gain access to a cloud host from inside a VM (e.g. [108],[109]) As cloud computing and especially IaaS relies heavily upon virtualization technologies this poses a threat to every cloud provider. By accessing the underlying host, an attacker not only gains access to real hardware but also to all other VMs running on this machine. It is therefore essential that VM escaping attacks get detected and appropriate counter measures are taken. This includes monitoring process activity on all cloud hosts for unusual activity, for example the commands that are being executed or syscalls as described by Hofmeyr et al. in [208]. By applying SAaaS agents to the cloud hosts, running processes, user access or even changing system behaviour can be profiled and monitored. Thus, attacks can be detected by the SAaaS agents.

8.3.7 Detection of Cloud Resource Misuse

The possibility to quickly aggregate massive loads of computing power is very attractive, also for criminals. Technically, there is however no non-intrusive way to decide for what exactly cloud computing power is used. Even if there was, data protection laws and credibility concerns prevent such monitoring. One possibility to encounter misuse, is to thoroughly check credit card or other payment data as most criminals wouldn't want to pay for themselves if other means such as stolen credit cards exist. However, the cloud's computing power can also be used to carry out attacks on other targets. One possibility is to aggregate many VMs and use them to DDoS a single target and thereby preventing others to use its services. To detect such attacks, the

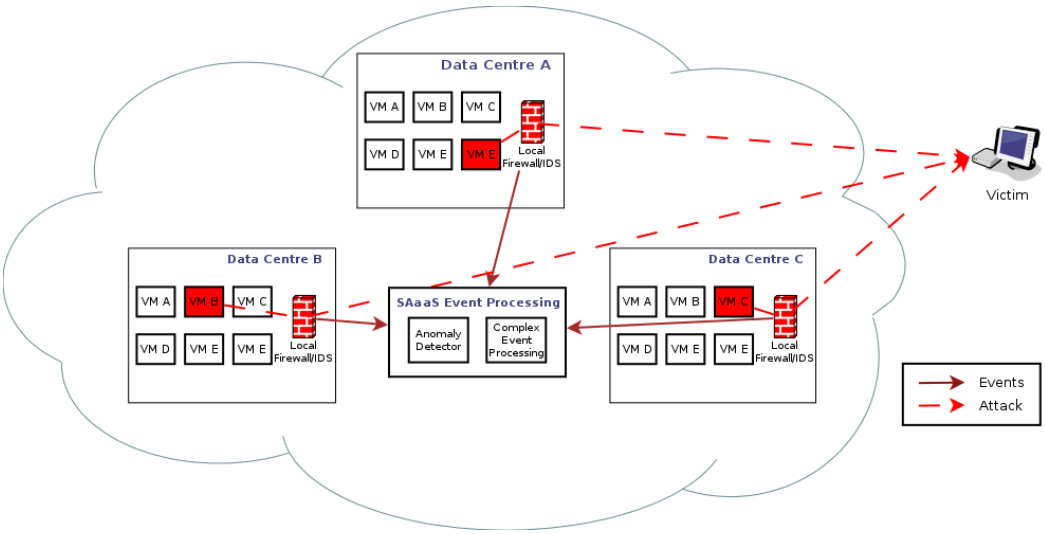


Figure 8.1: Distributed Denial of Service attack from the Cloud

network has to be monitored for anomalous activity especially from the inside. Due to the distributed nature of cloud computing information about network flow has to be collected at many different physical locations. To get the whole picture however, this data has to be analysed in the overall context.

An example scenario is depicted in figure 8.1. An attacker creates only a few number of instances in each of the cloud's data centers. Then he uses all of these world wide distributed instances to DDoS a victim's host. This host can be anywhere on the internet. If only the data in each data center was analysed for its own, maybe no attack would be detected as only a few number of hosts take part in it. If however, send the data to a centralized/distributed detection system, we can detect the attack at its whole extent.

8.4 Summary

In this chapter, it was elaborated how the developed concurrent audit system Security Audit as a Service addresses cloud specific security challenges, and that cloud audits can mitigate these problems. It was shown, how SAaaS improves cloud audits. The

interest of other companies on the developed Cloud Audit Test Criteria catalogue proves the novelty and value of this research.

The results from the practical validation of the prototype have demonstrated SAaaS is operating as defined by the architecture. SAaaS has successfully achieved the objectives of improving security in cloud environments by increasing transparency for cloud user and provider. Through utilising an agent framework, the weakness of traditional systems with a frequently changing infrastructure are overcome by the strength of performing specific targeted, concurrent security audits.

9

Conclusion & Future Work

"The best way to predict the future is to invent it."

(Alan Kay, Computer Scientist)

This chapter concludes the thesis by summarising the achievements of the research presented. It also discusses limitations of the work presented and identifies areas of further research.

9.1 Achievements of the Research

Cloud Computing provides new potential in flexible usage of computing resources, including rapid deployment, rapid elasticity and on-demand usage and payment. Chapter 2 introduced the evolution of cloud computing and presented cloud deployment and service models, typical roles and related technologies. An overview over the current commercial landscape is given and advantages as well as disadvantages were discussed. To lay out the basis of this research, Chapter 3 evaluated the current situation of cloud computing security. Security issues amplified by cloud characteristics as well as cloud specific security problems were identified. A novel approach of "concurrent cloud audits" was elaborated, which addresses the identified cloud security issues. As a first result, a novel Cloud Audit Test Criteria Catalogue was developed. It consists of over 140 audit test criteria for cloud infrastructures. It provides a valid contribution to the area of applied cloud security, since it enables cloud customer as well as provider with a detailed check list to evaluate their cloud computing infrastructure against security requirements. To the best knowledge of the author, no similar catalogue existed by the time of publication. The catalogue is already used in industry, for example by the management of the data centre of the federal state Rhineland-Palatinate (Germany). Chapter 4 presents the novel cloud audit system "Security Audit as a Service", which reacts with specific security audits on changes within a cloud infrastructure. Therefore, new software agents were developed to interact with cloud components and detect security incidents. It was acknowledged that this technique must be capable of providing the requirements for concurrent security audits of cloud components. A technical evaluation showed, that SAaaS agents are lightweight enough to fulfil the requirements of concurrent security cloud audits. The SAaaS architecture forms the second main contribution of this research after the cloud audit test criteria catalogue .

To be able to model desired security states and the proposed cloud security audits, a

novel Cloud Audit Policy Language (CAPL) was developed, and presented in Chapter 5. An evaluation of existing security policy languages identified the need for the new policiy language, since none of them respects cloud specific characteristics. Therefore, the developed Cloud Audit Policy Language is based on the Cloud Infrastructure and Management Interface Standard. Its novel contribution is the transformation of cloud security policies in corresponding software agents, monitoring cloud components and performing specific security audits in case of a detected infrastructure change. The developed Cloud Audit Policy Language is seamlessly integrated into the SAaaS architecture, providing a connection between modelled security states and corresponding cloud audits and the SAaaS agents. The Cloud Audit Policy Language forms the third main contribution of this research.

To address advanced cloud security issues as well as future cloud security attacks, anomaly detection in IaaS clouds based on behaviour analysis is proposed in Chapter 6. It is identified, that two classes of detection methods exist: rule based detection and behavioural based anomaly detection. It is shown, that the developed SAaaS system and the Cloud Audit Policy Language are supporting rule based anomaly detection. To address behavioural anomaly detection, an approach utilising artificial neural networks was presented. Due to absence of a publicly available cloud usage data set, a cloud usage simulator was developed, which was utilised to evaluate the feasibility of the neural network approach. It has been shown, that neural networks can be a valid approach to address more complex and unknown cloud security issues.

After presenting the design of the Security Audit as a Service architecture, a prototype was developed and presented in Chapter 7. In three development stages, representing the developed SAaaS agent architecture, the integration of the Cloud Audit Policy Language and the anomaly detection system, the feasibility of the system is presented. Three SAaaS demos demonstrate, that the developed SAaaS system fulfils the requirement of concurrent cloud audits, established in Chapter 3.

Chapter 8 provides a qualitative evaluation of the Security Audit as a Service architecture. It shows, how the SAaaS system addresses the established security issues and improves cloud audits. SAaaS enhances cloud security by increasing transparency in cloud environments and mitigating the risks of: *"Cloud Resource Abuse"*, *"Shared Technology Issues"*, *"Account Misuse"* and detection of several cloud specific attacks.

The research has met all of the objectives originally outlined in Chapter 1 and has resulted in the design and development of an Security Audit as a Services system. In addition, a number of papers relating to the research programme have been published in internationally recognised journals and presented at refereed conferences. In particular, the author has been awarded with a best paper prize at the International Conference on Emerging Network Intelligence (EMERGING 2011), an annual best paper prize from the HMD publisher, and the second place at the SERVICES CUP 2012 at the 8th IEEE World Congress on Services (SERVICES2012). Aspects of the research are already utilised by security provider in Germany utilising the Cloud Audit Test Criteria catalogue to strengthen cloud computing audits.

9.2 Limitations of the Research

Although the objectives of the research programme have been met, a number of decisions had to be made, which imposed limitations upon the work. The main limitations of the research are summarised below:

Privacy Concerns. The presented SAaaS agents are sending events, containing customer specific data such as usage data, VM utilisation or audit results to a central archive. This research assumes, that the cloud customer trusts the cloud provider to handle this data responsibly. However, this introduces a privacy issue, which is not acceptable for a real world implementation.

Limitations of SAaaS agents. The developed SAaaS sensor agents are monitoring cloud components for certain changes. However, no hardening of the agents, nor an agent integrity validation system was implemented, monitoring possible tampering of agents. With the prototype presented, it is possible for an attacker to

- Re-engineer an agents source code from the binary
- Replace an agent during runtime with a customised version
- Overload the system by event flooding, causing the manager agent to fail

These limitations were accepted during the design phase. However, on the last one, the JADE developer was contacted to fix this behaviour.

Cloud Audit Policy Language. The Cloud Audit Policy Language was developed, since evaluation showed, that no existing security policy language was fulfilling the established SAaaS requirements. Although CAPL was developed close to the CIMI standard, its application remains so far solely to this research. No other implementation, nor a vital user community exists. The connection between modelled security policies and corresponding SAaaS agent configuration was demonstrated for the introduced use cases. However, for an industry wide application a vital user community similar to the Nagios community would be necessary to create more CAPL plugins.

Anomaly Detection Simulation Data. A number of practical limitations exist with the anomaly detection based on cloud usage data. Since no read data set was available the developed simulator creates a simplified cloud usage data set. Especially, the simulated anomalies are limited in their occurrence and sort of type. Insufficient time and resources were available to further assess the integration of more input features for neural networks. Given the results from the simulations presented, it it is expected to achieve even more insights and benefits from the machine learning approach in cloud misusage cases.

Despite these limitations, the research programme has made valid contributions to knowledge and provided sufficient proof of concept for the ideas proposed.

9.3 Suggestions & Scope for Future Work

This research program has advanced the field of cloud computing security and cloud audits. However, a number of areas of scope for future work exist, specifically related to this research and more generally within the area of cloud misusage detection. These suggestions are detailed below:

Wider deployment. A wider and practical deployment of SAaaS at a public cloud provider would permit a comprehensive and thorough evaluation of the technique with real cloud usage data and customer demands.

Neural Networks vs. Statistical Approaches. The work on anomaly detection in clouds presented in this research is only a part of the overall research on Security Audit Compliance For Cloud Computing. Its task was to prove, that by analysing cloud usage behaviour anomaly detection in IaaS Cloud is possible. This was successfully done, as presented in Chapter 6. However, this just opens up a wide field of additional research topics. A comprehensive comparison of other anomaly detection methods, such as outlier detection or statistical approaches would give more insights into their applicability to cloud environments.

Automatic Identification of Input parameters. The presented anomaly detection system utilising neural networks uses manually identified input parameters. However, it remains unclear, how valuable they are for detection of other cloud attacks than presented. To cover a broader spectrum of cloud security issues, an anomaly detection system would need to automatically evaluate quality of input parameters. Two approaches are imaginable to get around this problem: a) A prior clustering algorithm module for each classifier which identifies valuable boundaries of it, repre-

senting normal behaviour b) User get the possibility to tag events if they are normal or abnormal (same technique used to improve spam filters)
This would have the following implications: anomalies, which can be detected by a certain feature from a statistical arithmetic average of a certain classifier get detected by the `Classifier Coordination` module. The module serves furthermore for the `Anomaly Detection` module to label misuse cases for the corresponding input parameters. The Anomaly Detection module (utilising for example neural nets) is used to learn not obvious connections of input classifiers and is well suited for anomaly detection in scenarios with increased complexity.

9.4 The Future of Cloud Computing Security

Cloud Computing will continuously grow, successfully changing our way we utilise IT resources. It leads to a more flexible way of accessing, using and releasing computing power and peripherals in a truly on-demand fashion. The in parallel strongly increasing mobile computing area, develops more and more power full tablets and smart phones, which form the access devices to cloud services, providing information at our finger tips. However, this also raises the demand for trust and security in cloud technology. Particularly, the current debate about disclosed information on spying and interception of information by national agencies [240] shows, that transparency and well established security technology is necessary to secure this new flexibility. It can be argued at this stage that cloud security systems no longer becomes an option but a necessity, for the so far more reserved information policy of cloud providers. Although Cloud provider realised this development and are slowly extending their security measures, this research has highlighted the need to provide increased security mechanisms tailored to the specific characteristics a cloud architecture introduces. The research has, through a comprehensive analysis, designed and developed a flexible cloud audit

system capable of providing transparent and continuous information about the security status of cloud instances. Thus, security is no longer an one-off process but a continual confidence based measure, capable of utilising a wide range of information techniques to maintain systems confidentiality and integrity.

In conclusion, security will become more important and will be a decision criterion for enterprises moving services into cloud computing technology. The ability to have a clear understanding about cloud instances, their location and the security state of surrounding infrastructure will be fundamental to the successful deployment of future cloud services.

List of Publications

The following lists published work on the presented research of Cloud computing security by the author during the PhD. It is listed in categories of publication types and within those in chronological order.

Journal Papers

1. Frank Doelitzscher, Anthony Sulistio, Christoph Reich, Hendrik Kuijs and David Wolf, "**Private Cloud for Collaboration and e-Learning Services: from IaaS to SaaS**", Journal: Computing, ISSN: 0010-485X, DOI: 10.1007/s00607-010-0106-z, Springer, July 2010

2. Frank Doelitzscher, Mathias Ardelt, Martin Knahl, Christoph Reich, "**Security issues in IT outsourcing through cloud computing**", HMD - Praxis der Wirtschaftsinformatik", Volume 281, ISSN 1436-3011, Oct. 2011
   Winner of Annual Best Paper Prize

3. Frank Doelitzscher, Christoph Reich, Martin Knahl and Nathan Clarke, "**An agent based business aware incident detection system for cloud environments**", Journal of Cloud Computing: Advances, Systems and Applications, 1:9, ISSN: 2192-113X, DOI: 10.1186/2192-113X-1-9, SpringerOpen, July 2012

4. Frank Doelitzscher, Thomas Ruebsamen, Tina Karbe, Christoph Reich, Nathan Clarke, "**Sun Behind Clouds - On Automatic Cloud Security Audits and a Cloud Audit Policy Language**", International Journal On Advances in Networks and Services, Volume 6 Nr. 1&2 2013, July 2013

Conference Papers

5. Anthony Sulistio, Christoph Reich and Frank Doelitzscher, "**Cloud Infrastructure & Applications - CloudIA**", Proceedings of the 1st International Conference on Cloud Computing (CloudCom 2009), China, Dec. 1-4, 2009, (Short Paper)

6. Frank Doelitzscher, Christoph Reich and Anthony Sulistio, "**Designing Cloud Services Adhering to Government Privacy Laws**", Proceedings of the 3rd IEEE International Symposium on Trust, Security and Privacy for Emerging Applications (TSP-10), 978-1-4244-7547-6 pp930-935, DOI: 10.1109/CIT.2010.172, Bradford, June 29-July 1, 2010

7. Frank Doelitzscher and Christoph Reich and Martin Knahl and Nathan Clarke, "**Incident detection for cloud environments**", Proceedings of the Third International Conference on Emerging Network Intelligence (EMERGING 2011), ISBN: 978-1-61208-174-8, pp100-105, Lisboa, Nov. 20-25, 2011
Winner of Conference Best Paper Prize

8. Frank Doelitzscher, Christoph Reich, Martin Knahl and Nathan Clarke, "**An autonomous agent based incident detection system for cloud environments**", Proceedings of the 3rd IEEE International Conference on Cloud Computing Technology and Science (CloudCom2011), ISBN: 978-0-7695-4622-3/11, pp197-204, DOI:10.1109/CloudCom.2011.35, Athens, Nov. 29th-Dec. 1st, 2011

9. Frank Doelitzscher, Anthony Sulistio, Markus Held and Christoph Reich, "**ViteraaS: Virtual Cluster as a Service**", Proceedings of the 3rd IEEE International Conference on Cloud Computing Technology and Science (CloudCom2011), ISBN: 978-0-7695-4622-3/11, pp652-657, DOI: 10.1109/CloudCom.2011.101, Athens, Nov. 29th-Dec. 1st, 2011

10. Frank Doelitzscher, Christian Fischer, Denis Moskal, Christoph Reich, Martin Knahl, and Nathan Clarke, "**Validating Cloud Infrastructure Changes by Cloud Audits**", Proceedings of the 8th IEEE World Congress on Services (SERVICES2012), ISBN:978-0-7695-4756-5/12, DOI: 10.1109/SERVICES.2012.12, Honolulu, Jun 24th-29th, 2012
Winner 2nd place of IEEE SERVICES CUP 2012

11. Frank Doelitzscher, Christoph Reich, Martin Knahl, and Nathan Clarke, "**Anomaly Detection In IaaS Clouds**", 5th IEEE International Conference on Cloud Computing Technology and Science (CloudCom2013), Bristol, Dec 02th-05th, 2013

Contribution to edited books

12. Anthony Sulistio, Frank Doelitzscher and Christoph Reich, "**Automated Virtual Machine Creation with On-Demand Software Installation**", book chapter- Computer Science Research and Technology. Volume 3, ISBN: 978-1-61122-074-2 Nova Publishers, 2011

13. Frank Doelitzscher, Christoph Reich, Martin Knahl and Nathan Clarke, "**Understanding Cloud Audits**", book chapter: Privacy and Security for Cloud Computing, ISBN: 978-1-4471-4188-4, Springer, Aug 2012

Research magazines

14. Frank Doelitzscher and Christoph Reich, "**Cloud-Computing im Rechenzentrum der Hochschule Furtwangen**", OBJECT spektrum - Volume 06/2009, Sep. 2009

15. Frank Doelitzscher and Christoph Reich, "**Audit criteria for cloud computing environments**", Hackin9 EXTRA, Volume 4/2012, October 2012

Bibliography

[1] T. Mather, S. Kumaraswamy, and S. Latif, *Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance*. O'Reilly Media, Inc., 2009.

[2] Gartner Inc. (2010) Gartner's 2010 Hype Cycle Special Report Evaluates Maturity of 1,800 Technologies. [Online]. Available: http://www.gartner.com/it/page.jsp?id=1447613-Accessed:10.10.2012

[3] Gartner. (2011) Hype Cycle for Cloud Computing 2011. [Online]. Available: http://softwarestrategiesblog.com/2011/07/27/gartner-releases-their-hype-cycle-for-cloud-computing-2011-Accessed: 31.10.2012

[4] Experton Group. (2010) Investments for Cloud Computing. [Online]. Available: http://de.statista.com/statistik/daten/studie/172478/umfrage/prognostizierte-investitionen-und-ausgaben-fuer-cloud-computing-bis-2015-Accessed: 31.10.2013

[5] Pierre Audoin Consultants. (2010) D3 - Baseline Scenario for 2020: Economic and Social Impact of Software & Software-Based Services. [Online]. Available: http://de.statista.com/statistik/daten/studie/169271/umfrage/prognostizierter-umsatz-mit-cloud-computing-weltweit-seit-2008-Accessed: 31.10.2012

[6] EMC, "THE CLOUD DIVIDEND: Part One The economic benefits of cloud computing to business and the wider EMEA economy France, Germany, Italy, Spain and the UK," Tech. Rep., 2010. [Online]. Available: http://uk.emc.com/microsites/2011/cloud-dividend/index.htm-Accessed:31.10.2013

[7] B. Grobauer, T. Walloschek, and E. Stocker, "Understanding cloud computing vulnerabilities," *Security Privacy, IEEE*, vol. 9, no. 2, pp. 50 –57, march-april 2011.

[8] Forbes Insights, "Seeding the Cloud - Enterprises Set Their Strategies for Cloud Computing," Tech. Rep., 2010.

[9] F. Doelitzscher, C. Reich, T. Hartmann and M. Schaefer, "Audit test criteria catalouge for cloud computing environments," University of Applied Sciences Furtwangen, Tech. Rep., 2012. [Online]. Available: http://wolke.hs-furtwangen.de/assets/files/AuditPruefkriterienkatalogCloudComputing.pdf-Accessed:31.10.2013

[10] F. Doelitzscher, C. Reich, T. Hartmann, and M. Schaefer, "Audit test criteria catalouge - electronic criteria spreadsheet," University of Applied Sciences Furtwangen, Tech. Rep., 2012. [Online]. Available: http://wolke.hs-furtwangen.de/assets/files/Anhang1_Pruefkriterien_Checkliste_Version1.xls

[11] D. B. Lange, "Mobile objects and mobile agents: The future of distributed computing?" in *In Proceedings of The European Conference on Object-Oriented Programming '98*, 1998, pp. 1–12.

[12] James A. Hoagland, "Specifying and Implementing Security Policies Using LaSCO, the Language for Security Constraints on Objects," Ph.D. dissertation, University of California Davis, 2000. [Online]. Available: http://seclab.cs.ucdavis.edu/projects/arpa/LaSCO/dis/dissertation.pdf

[13] C. Reich, S. Hbner, and H. Kuijs, "Cloud computing for on-demand virtual desktops and labs," *Cloud Computing for Teaching and Learning: Strategies for Design and Implementation*, pp. 111–125, 2013.

[14] J. Spring, "Monitoring cloud computing by layer, part 1," *Security Privacy, IEEE*, vol. 9, no. 2, pp. 66 –68, 2011.

[15] Matthias Luft. (2013, 05) Analysis of hypervisor breakouts. [Online]. Available: http://www.insinuator.net/2013/05/analysis-of-hypervisor-breakouts/#more-2167-Accessed:01.07.2013

[16] M. Lundin, "Industry issues and standards - effectively addressing compliance requirements," *ISACA San Francisco Chapter, Consumer Information Protection Event*, 04 2009. [Online]. Available: http://mscerts.programming4.us/programming/cloud%20security%20and%20privacy%20%20%20regulatoryexternal%20compliance%20%28part%201%29.aspx-Accessed:21.08.2013

[17] Navint Inc., "Trends in cloud adoption 2012," Tech. Rep., 2012.

[18] Dropbox Inc. (2012) dropbox. [Online]. Available: https://www.dropbox.com-Accessed:31.10.2013

[19] ownCloud Inc. ownCloud. [Online]. Available: http://owncloud.org-Accessed:31.10.2013

[20] Apple Inc. iCloud. [Online]. Available: https://www.icloud.com-Accessed: 31.10.2013

[21] Microsoft. (2013) Microsoft Office Online Services 365. [Online]. Available: http://office365.microsoft.com-Accessed:31.10.2013

[22] Google Inc. (2012) Google Apps. [Online]. Available: http://www.google.com/apps-Accessed:31.10.2013

[23] Amazon. (2012) Amazon Web Services (AWS). [Online]. Available: http://aws.amazon.com-Accessed:31.10.2013

[24] A. W. Services. (2012, 04) SIMPLE MONTHLY CALCULATOR. [Online]. Available: http://calculator.s3.amazonaws.com/calc5.html?lng=de_DE-Accessed:31.10.2013

[25] Gerry Grealish. Healthcare Data on the Cloud - The Reality of Sensitive Information Online. [Online]. Available: http://www.perspecsys.com/healthcare-data-on-the-cloud-the-reality-of-sensitive-information-online-Accessed: 31.10.2013

[26] Cloud Security Alliance, "Top Threats to Cloud Computing V1.0," 2010. [Online]. Available: https://cloudsecurityalliance.org/topthreats.html-Accessed: 31.10.2013

[27] Anthony Sulistio and Christoph Reich and Frank Doelitzscher, "Cloud Infrastructure & Applications - CloudIA," in *Proceedings of the 1st International Conference on Cloud Computing (CloudCom'09)*, Beijing, China, 12 2009.

[28] N. G. Carr, *The Big Switch. Der grosse Wandel. Die Vernetzung der Welt von Edison bis Google*, 1st ed. Heidelberg: Mitp, 2009. [Online]. Available: http://d-nb.info/991171578/04

[29] Gartner Research Group. (2012) Hype Cycle Research Methodology. [Online]. Available: http://www.gartner.com/technology/research/methodologies/hype-cycle.jsp-Accessed:31.10.2013

[30] CIO.de. (2013) The new gartner hype cycle 2013. [Online]. Available: www.cio.de/strategien/2927010-Accessed:09.10.2013

[31] M. Vouk, S. Averritt, M. Bugaev, A. Kurth, A. Peeler, H. Schaffer, E. Sills, S. Stein, and J. Thompson, ""Powered by VCL" – Using Virtual Computing Laboratory (VCL) Technology to Power Cloud Computing," in *Proceedings of the 2nd International Conference on the Virtual Computing Initiative (ICVCI'08)*, May 16–17 2008.

[32] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud Computing and Grid Computing 360-Degree Compared," in *Proceedings of the Grid Computing Environments Workshop (GCE'08)*, Austin, Texas, USA, Nov. 16 2008.

[33] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing," Univ. of California at Berkeley, Tech. Rep. UCB/EECS-2009-28, 2009. [Online]. Available: http://berkeleyclouds.blogspot.com/2009/02/above-clouds-released.html

[34] P Mell, T Grance, "Effectively and securely using the cloud computing paradigm," US National Institute of Standards and Technology, Tech. Rep., 2009. [Online]. Available: http://csrc.nist.gov/groups/SNS/cloud-computing-Accessed:31.10.2013

[35] P. Mell and T. Grance, "The nist definition of cloud computing," *National Institute of Standards and Technology*, vol. 53, no. 6, p. 50, 2009. [Online]. Available: http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc-Accessed:31.10.2013

[36] UK Government. (2012) The G-Cloud Programme. [Online]. Available: http://gcloud.civilservice.gov.uk-Accessed:31.10.2013

[37] Trend Micro Inc. (2012) Smart Protection Network. [Online]. Available: http://www.trendmicro.com/us/technology-innovation/our-technology/smart-protection-network-Accessed:31.10.2013

[38] KVM - The Linux Kernel-Based Virtual Machine. How you can use qemu/kvm base images to be more productive. [Online]. Available: http://www.linux-kvm.com/content/how-you-can-use-qemukvm-base-images-be-more-productive-part-1-Accessed:01.10.2013

[39] Xen.org. (2011, 04) Xen Hypervisor. [Online]. Available: http://www.xen.org/products/xenhyp.html-Accessed:10.10.13

[40] VMware. VMWare Hypervisor. [Online]. Available: http://www.vmware.com/support/vsphere-hypervisor.html-Accessed:10.10.2013

[41] B. Halpert, *Auditing Cloud Computing: A Security and Privacy Guide*, ser. Wiley Corporate F&a. John Wiley & Sons, 2011.

[42] Rackspace.com. (2012) Rackspace.com. [Online]. Available: http://www.rackspace.com-Accessed:31.10.2013

[43] Hosting.com. (2012) Hosting.com. [Online]. Available: http://www.hosting.com-Accessed:31.10.2013

[44] Salesforce.com Inc. (2012) Force.com. [Online]. Available: http://www.force.com-Accessed:31.10.2013

[45] Microsoft Inc. (2013) Windows Azure. [Online]. Available: www.microsoft.com/en-us/cloud/default.aspx-Accessed:31.10.2012

[46] Google Inc. (2012) Google Docs. [Online]. Available: http://www.google.com/docs-Accessed:31.10.2013

[47] Salesforce.com Inc. (2012) Salesforce.com. [Online]. Available: http://www.force.com-Accessed:31.10.2013

[48] L. Youseff, M. Butrico, and D. Da Silva, "Toward a Unified Ontology of Cloud Computing," in *Grid Computing Environments Workshop, 2008. GCE '08*, 2008, pp. 1 –10.

[49] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of Internet Services and Applications*, vol. 1, pp. 7–18, 2010, 10.1007/s13174-010-0007-6. [Online]. Available: http://dx.doi.org/10.1007/s13174-010-0007-6

[50] A. Sinha, "Client-server computing," *Commun. ACM*, vol. 35, no. 7, pp. 77–98, Jul. 1992. [Online]. Available: http://doi.acm.org/10.1145/129902.129908

[51] I. Foster, C. Kesselman, and S. Tuecke, "The anatomy of the grid: Enabling scalable virtual organizations," *International Journal of High Performance Computing Applications*, vol. 15, no. 3, pp. 200–222, Fall 2001. [Online]. Available: http://www2002.org/foster.pdf

[52] F. Doelitzscher, M. Held, C. Reich, and A. Sulistio, "Viteraas: Virtual cluster as a service," in *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*, 29 2011-dec. 1 2011, pp. 652 –657.

[53] P. Laird. (2009) Visual map of the cloud computing/saas/paas markets: September 2008 update. [Online]. Available: http://peterlaird.blogspot.de/2008/09/visual-map-of-cloud-computingsaaspaas.html-Accessed:31.10.2013

[54] OpenCrowd. (2010) Cloud Taxonomy. [Online]. Available: http://www.opencrowd.com/assets/images/views/views_cloud-tax-lrg.png-Accessed:31.10.2013

[55] SearchCloudComputing.com. Top 10 cloud computing providers of 2011. [Online]. Available: http://searchcloudcomputing.techtarget.com/feature/Top-10-cloud-computing-providers-of-2011-Accessed:31.10.2013

[56] Verizon /Terremark. (2012) vCloud Express. [Online]. Available: http://vcloudexpress.terremark.com-Accessed:31.10.2013

[57] IBM. (2012) Smart Business Test and Development Cloud. [Online]. Available: http://www-935.ibm.com/services/de/de/it-services/smart-business-development-und-test-cloud.html-Accessed:31.10.2013

[58] Computer Sciences Corporation. (2012) BizCloud. [Online]. Available: http://www.csc.com/cloud/offerings/53410/59294-csc_bizcloud-Accessed:31.10.2013

[59] OpenStack. (2012) Open Source Cloud Computing Software. [Online]. Available: http://www.openstack.org-Accessed:31.10.2013

[60] BlueLock Inc. (2012). [Online]. Available: http://www.bluelock.com-Accessed: 31.10.2013

[61] Joyent Inc. (2012). [Online]. Available: http://www.joyent.com-Accessed: 01.10.2013

[62] Amazon Web Services. (2009) AF83 Case Study. [Online]. Available: http://aws.amazon.com/solutions/case-studies/af83-Accessed:31.10.2010

[63] Amazon WebServices. (2012) Amazon S3 Organizer(S3Fox). [Online]. Available: https://addons.mozilla.org/en-US/firefox/addon/amazon-s3-organizers3fox-Accessed:18.08.2013

[64] Hetzner. (2012, 04) Root Server X4. [Online]. Available: http://www.hetzner.de/en/hosting/produkte_rootserver/x4-Accessed:31.10.2013

[65] Gartner Inc. (2013) Magic Quadrant for Cloud Infrastructure as a Services. [Online]. Available: http://www.gartner.com/technology/reprints.do?id=1-1IMDMZ5&ct=130819&st=sb-Accessed:08.10.2013

[66] U.S. Department of Health and Human Services (DHHS), "Health Insurance Portability and Accountability Act (HIPAA)," 1996.

[67] 107th Congress (2001-2002), "H.R.3763 – Sarbanes-Oxley Act of 2002," 2002. [Online]. Available: http://thomas.loc.gov/cgi-bin/query/z?c107:H.R.3763.ENR:-Accessed10.10.2013

[68] Payment Card Industry Security Standards Council, "Payment Card Industry Data Security Standard," 2010. [Online]. Available: https://de.pcisecuritystandards.org/_onelink_/pcisecurity/en2de/minisite/en/docs/pci_dss_v2-0.pdf-Accessed:10.10.2013

[69] American Institute of Certified Public Accountants (AICPA). Statement on Auditing Standards (SAS) No. 70. [Online]. Available: http://sas70.com/sas70_overview.html-Accessed10.10.2013

[70] F. Dölitzscher, C. Reich, and A. Sulistio, "Designing cloud services adhering to government privacy laws," in *Proceedings of 10th IEEE International Conference on Computer and Information Technology (CIT 2010)*, 2010, pp. 930–935.

[71] F. Doelitzscher, M. Ardelt, M. Knahl, and C. Reich, "Sicherheitsprobleme fr it outsourcing basierend auf cloud computing," *HMD - Praxis der Wirtschaftsinformatik*, vol. 281, pp. 62–70, 10 2011.

[72] F. Doelitzscher, C. Reich, M. Knahl, and N. Clarke, *Understanding Cloud Audits*, ser. Computer Communications and Networks. Springer, 2013, ch. 4, pp. 125–163.

[73] Cloud Security Alliance, "Security Guidance for Critical Areas of Focus in Cloud Computing v2.1," 2009.

[74] European Network and Information Security Agency, "Cloud Computing Security Risk Assessment," European Network and Information Security Agency, Tech. Rep., 2009.

[75] L. Vaquero, L. Rodero-Merino, and D. Moran, "Locking the sky: a survey on iaas cloud security," *Computing*, vol. 91, pp. 93–118, 2011.

[76] Y. Chen, V. Paxson, and R. H. Katz, "What's New About Cloud Computing Security?" EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2010-5, 01 2010.

[77] The Washington Post. (2007) Salesforce.com acknowledges data loss. [Online]. Available: http://voices.washingtonpost.com/securityfix/2007/11/salesforcecom_acknowledges_dat.html-Accessed:10.10.2013

[78] ——. (2009) Sidekick users see their data vanish into a cloud. [Online]. Available: http://articles.washingtonpost.com/2009-10-11/news/36921791_1_sidekick-owners-sidekick-users-personal-content-Accessed:10.10.2013

[79] Security Focus. Zeus botnet finds hold in Amazon cloud. [Online]. Available: http://www.securityfocus.com/brief/1046-Accessed:24.06.2013

[80] Business_Insider, "Inside Amazon's Cloud Disaster," http://www.businessinsider.com/amazon-outage-enters-its-second-day-lots-of-sites-still-down-2011-4, 2011.

[81] Business_Insider. (2011) Amazon's Cloud Crash Disaster Permanently Destroyed Many Customers' Data. [Online]. Available: http://www.businessinsider.com/amazon-lost-data-2011-4-Accessed:31.10.2012

[82] WhiteC0de.com. (2012) New hotmail exploit can get any hotmail email account hacked for just 20$. [Online]. Available: http://www.whitec0de.com/new-hotmail-exploit-can-get-any-hotmail-email-account-hacked-for-just-20-Accessed:10.10.2013

[83] International Organisation on Standardization, "Iso 27001:2005," http://iso.org, 2010.

[84] *ISO 9001:2008 Qauality Managemt Systems - Requirements*, International Organization for Standardization Std.

[85] Amazon Web Services. (2011) Amazon Service Health Dashboard. [Online]. Available: http://status.aws.amazon.com-Accessed:18.08.2013

[86] Cloud Security Alliance, "Security Guidance for Critical Areas of Focus in Cloud Computing v3.0," 2011.

[87] ——, "The Notorious Nine - Cloud Computing Top Threats in 2013," 2013. [Online]. Available: https://cloudsecurityalliance.org/topthreats.html-Accessed: 31.10.2013

[88] Federal Office for Information Security Germany, "Security Recommendations for Cloud Computing Provider - original title: Sicherheitsempfehlungen fr Cloud Computing Anbieter," Tech. Rep., 2011. [Online]. Available: https://www.bsi.bund.de/DE/Themen/CloudComputing/Eckpunktepapier/Eckpunktepapier_node.html-Accessed:31.10.2013

[89] BSI, "Security recommendations for cloud computing provider - Original: Sicherheitsempfehlungen fr Cloud Computing Anbieter," Bundesamt fuer Sicherheit in der Informationstechnik, Tech. Rep., 2012.

[90] S. Pearson, "Taking Account of Privacy when Designing Cloud Computing Services," in *Proceedings of the ICSE Workshop on Software Engineering Challenges of Cloud Computing*, Vancouver, Canada, May 23 2009.

[91] M. Balduzzi, J. Zaddach, D. Balzarotti, E. Kirda, and S. Loureiro, "A security analysis of amazon's elastic compute cloud service," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, ser. SAC '12. New York, NY, USA: ACM, 2012, pp. 1427–1434.

[92] Cloud Security Alliance. (2011) Governance, Risk Management and Compliance (CSR). [Online]. Available: https://cloudsecurityalliance.org/research/grc-stack-Accessed:31.10.2013

[93] C. Hoss, S. Johnston, G. Reese, and B. Sapiro, "CloudAudit 1.0 - Automated Audit, Assertion, Assessment, and Assurance API (A6)," Internet Engineering Task Force - Internet Draft - Experimental, Tech. Rep., 2010. [Online]. Available: https://tools.ietf.org/html/draft-hoff-cloudaudit-00-Accessed:31.10.2013

[94] Cloud Security Alliance. (2011) CloudTrust Protocol Information Overview Powerpoint. [Online]. Available: https://cloudsecurityalliance.org/research/ctp-Accessed:31.10.2013

[95] EuroCloud Deutschland_eco e.V. (2011) Eurocloud star audit saas certificate. [Online]. Available: http://www.saas-audit.de-Accessed:31.10.2013

[96] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," in *INFOCOM, 2010 Proceedings IEEE*, march 2010, pp. 1 –9.

[97] Y. Zhu, G. Ahn, H. Hu, S. Yau, H. An, and S. Chen, "Dynamic Audit Services for Outsourced Storages in Clouds," *Services Computing, IEEE Transactions on*, vol. PP, no. 99, p. 1, 2011.

[98] P. Massonet, S. Naqvi, C. Ponsard, J. Latanicki, B. Rochwerger, and M. Villari, "A Monitoring and Audit Logging Architecture for Data Location Compliance in Federated Cloud Infrastructures," in *Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), 2011 IEEE International Symposium on*, may 2011, pp. 1510 –1517.

[99] D. Tancock, S. Pearson, and A. Charlesworth, "A privacy impact assessment tool for cloud computing," in *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, 2010, pp. 667–676.

[100] L. Li, L. Xu, J. Li, and C. Zhang, "Study on the third-party audit in cloud storage service," in *Cloud and Service Computing (CSC), 2011 International Conference on*, dec. 2011, pp. 220 –227.

[101] Federal Office for Information Security, "IT-Grundschutz Catalogues," Tech. Rep., 2011. [Online]. Available: https://www.bsi.bund.de/EN/Topics/ITGrundschutz/itgrundschutz_node.html-Accessed20.08.2013

[102] National Institute of Standards and Technology (NIST), "Engineering Principles for Information Technology Security," NIST, Tech. Rep., 2004.

[103] L. J. Sotto, B. C. Treacy, and M. L. McLellan, "Privacy and Data Security Risks in Cloud Computing," in *Electronic Commerce & Law Report*, 02 2010.

[104] Liebermann Software, "2011 Survey of IT Professionals Password Practices and Outcomes," Tech. Rep., 2011.

[105] Amazon Web Services. (2010) AWS Achieves PCI DSS Level 1 Compliance and ISO 27001 Certification. [Online]. Available: http://aws.amazon.com/security/#certifications-Accessed:31.10.2013

[106] Vmware command line tools. [Online]. Available: https://sites.google.com/site/chitchatvmback/vmtools-Accessed:01.07.2013

[107] J. Rutkowska. Our xen 0wning trilogy highlights. [Online]. Available: http://theinvisiblethings.blogspot.de/2008/08/our-xen-0wning-trilogy-highlights.html-Accessed:01.07.2013

[108] Kortchinsky, "Cloudburst," http://www.blackhat.com /presentations/bh-usa-09/KORTCHINSKY/BHUSA09-Kortchinsky-Cloudburst-PAPER.pdf, Tech. Rep., 2009.

[109] N. Elhage, "Virtunoid: A kvm guest → host privilege escalation exploit," in *Black Hat USA*, 2011. [Online]. Available: https://nelhage.com/talks/kvm-defcon-2011.pdf-Accessed:31.10.2013

[110] V. S. Advisory. Vmsa-2012-0009.2. [Online]. Available: http://www.vmware.com/security/advisories/VMSA-2012-0009.html-Accessed:18.08.2013

[111] M. Luft and P. Turbing, "Exploiting virtual file formats for fun and profit," ERNW GmbH, Tech. Rep., 2013.

[112] VUPEN Vulnerability Research Team. Advanced exploitation of xen hypervisor sysret vm escape vulnerability. [Online]. Available: http://www.vupen.com/blog/20120904.Advanced_Exploitation_of_Xen_Sysret_VM_Escape_CVE-2012-0217.php-Accessed:01.07.2013

[113] J. Rutkowska, "Xen 0wning Trilogy: code, demos and q35 attack details," http://theinvisiblethings.blogspot.com/2008/09/xen-0wning-trilogy-code-demos-and-q35.html, Tech. Rep., 2008.

[114] OpenNebula. (2012, July) OpenNebula Marketplace. [Online]. Available: http://marketplace.c12g.com/appliance-Accessed:10.06.2013

[115] Amazon. (2012, July) AWS Marketplace. [Online]. Available: http://aws.amazon.com/marketplace-Accessed:10.06.2013

[116] Amazon Web Services. (2012, July) How To Share and Use Public AMIs in A Secure Manner. [Online]. Available: http://aws.amazon.com/articles/0155828273219400-Accessed:10.06.2013

[117] S. Bugiel, S. Nürnberger, T. Pöppelmann, A.-R. Sadeghi, and T. Schneider, "Amazonia: when elasticity snaps back," in *Proceedings of the 18th ACM conference on Computer and communications security*, ser. CCS '11. New York, NY, USA: ACM, 2011, pp. 389–400.

[118] N. A. Haroon Meer. (2009) Clobbering the Cloud, part 4 of 5. [Online]. Available: http://www.defcon.org/images/defcon-17/dc-17-presentations/defcon-17-sensepost-clobbering_the_cloud.pdf-Accessed:10.06.2013

[119] German Parliament, *German Data Protection Act.* ISBN: 3406561632: Deutscher Taschenbuch Verlag, 2010.

[120] ComputerworldUK. (2011) Law enforcement agencies access rights to your cloud data. [Online]. Available: http://blogs.computerworlduk.com/cloud-vision/2011/07/law-enforcement-agencies-access-rights-to-your-cloud-data/index.htm-Accessed:31.10.2013

[121] E. H. Spafford and D. Zamboni, "Intrusion detection using autonomous agents," *Computer Networks*, vol. 34, no. 4, pp. 547 – 570, 2000, recent Advances in Intrusion Detection Systems.

[122] IT Infrastructure Library, http://www.itil-officialsite.com, 2012.

[123] Office of Government Commerce, *Service Operation Book (Itil)*. The Stationery Office, 2007, no. 978-0113310463.

[124] American Institute of Certified Public Accountants. (2012) The SSAE16 Auditing Standard. [Online]. Available: http://www.ssae-16.com-Accessed: 31.10.2013

[125] J. Sinclair, B. Hudzia, M. Lindner, H. Stewart, and T. Harmer, "Architecture for compliance analysis of distributed service based systems," 2011, pp. 286–292, iSSN: 978-989-8425-52-2.

[126] S. inclair. (2010) Auditing in cloud computing. SAP Research. [Online]. Available: http://de.slideshare.net/jonathansinclair86/cloud-auditing-Accessed: 18.08.2013

[127] A6 - Cloud Audit. (2011) Automated Audit, Assertion, Assessment, and Assurance. [Online]. Available: http://cloudaudit.org-Accessed:18.08.2013

[128] Cloud Security Alliance. (2011) Cloud security control matrix. [Online]. Available: https://cloudsecurityalliance.org/research/initiatives/cloud-controls-matrix-Accessed:31.10.2013

[129] EuroCloud Deutschland_eco e.V. (2011) Eurocloud quick reference. [Online]. Available: http://www.saas-audit.de/files/2011/04/110223-Quick_Reference_en.pdf-Accessed:31.10.2013

[130] ——. (2011) First firms certified with eurocloud star audit saas: optivo and pironet ndh datacentre. [Online]. Available: http://bit.ly/1atDkcM-Accessed: 31.10.2013

[131] eEye. (2011) Retina cloud. [Online]. Available: http://www.eeye.com/products/retina/cloud-Accessed:31.10.2011

[132] Secpoint. (2011) Cloud penetrator. [Online]. Available: http://www.secpoint.com/cloud-penetrator-web-vulnerability-scanner.html-Accessed:31.10.2013

[133] Kyplex. (2011) Website security and anti virus scanner. [Online]. Available: http://www.securitywizardry.com/index.php/products/scanning-products/website-scanners/kyplex-website-antivirus.html-Accessed:01.10.13

[134] F. Doelitzscher, C. Reich, M. Knahl, and N. Clarke, "An autonomous agent based incident detection system for cloud environments," in *Proceedings of 3rd IEEE International Conference on Cloud Computing Technology and Science (IEEE CloudCom 2011)*, 2011.

[135] N E Case. (2012, July) Extundelete. [Online]. Available: http://extundelete.sourceforge.net-Accessed:10.06.2013

[136] WinRecovery. (2012, July) Winundelete. [Online]. Available: http://www.winundelete.com-Accessed:10.06.2013

[137] German Cloud Provider. Toaster.NET GmbH. [Online]. Available: http://www.toasternet.eu-Accessed:11.10.2013

[138] Security Provider. SCHUTZWERK GmbH. [Online]. Available: http://www.schutzwerk.com-Accessed:11.10.2013

[139] Hochschule Furtwangen University. (2012) Security audit as a service project web site. [Online]. Available: http://www.wolke.hs-furtwangen.de/currentprojects/saaas-Accessed:28.08.2013

[140] Arbeitskreise Technik und Medien der Konferenz der Datenschutzbeauftragten des Bundes und der L"ander, "Orientierungshilfe - Cloud Computing," Tech. Rep., 2011.

[141] Rhineland-Palatinate web site. (2013) Rhineland-palatinate certifies as the first federal state the usage of a virtualized cloud infrastructure for a so called police cloud. [Online]. Available: http://bit.ly/HCeMIx-Accessed:31.10.2013

[142] F. Doelitzscher, C. Reich, M. Knahl, and N. Clarke, "An autonomous agent based incident detection system for cloud environments," in *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*, 29 2011-dec. 1 2011, pp. 197 –204.

[143] ——, "Incident detection for cloud environments," in *In Proceedings of International Conference on Emerging Network Intelligence (EMERGING 2011)*, no. ISBN: 978-1-61208-174-8, 2011, pp. 100–105, iSBN: 978-1-61208-174-8. [Online]. Available: http://www.thinkmind.org/index.php?view=article&articleid=emerging_2011_5_30_40134

[144] F. Doelitzscher, C. Fischer, D. Moskal, C. Reich, M. Knahl, and N. Clarke, "Validating cloud infrastructure changes by cloud audits," in *Services (SERVICES), 2012 IEEE Eighth World Congress on*, 2012.

[145] F. Doelitzscher, C. Reich, M. Knahl, A. Passfall, and N. Clarke, "An Agent Based Business Aware Incident Detection System for Cloud Environments," *Journal of Cloud Computing: Advances, Systems and Applications*, vol. 1, no. 1, p. 9, 2012.

[146] H. Raj and K. Schwan, "Extending virtualization services with trust guarantees via behavioral monitoring," in *Proceedings of the 1st EuroSys Workshop on Virtualization Technology for Dependable Systems*, ser. VDTS '09. New York, NY, USA: ACM, 2009, pp. 24–29.

[147] J. Wei, X. Zhang, G. Ammons, V. Bala, and P. Ning, "Managing security of virtual machine images in a cloud environment," in *Proceedings of the 2009 ACM workshop on Cloud computing security*, ser. CCSW '09. New York, NY, USA: ACM, 2009, pp. 91–96. [Online]. Available: http://doi.acm.org/10.1145/1655008.1655021

[148] Andreas Antonopoulos, "Securing Virtualized Infrastructure: From Static Security to Virtual Shields," Nemertes Research, Tech. Rep., 2007.

[149] Schaaf, M., Koschel, A., Grivas, S. G., Astrova, I., "An active DBMS style activity service for cloud environments," in *Cloud Computing 2010 : The First International Conference on Cloud Computing, GRIDs, and Virtualization*, 2010.

[150] T. Ries, V. Fusenig, C. Vilbois, and T. Engel, "Verification of data location in cloud networking," in *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on*, 2011, pp. 439–444.

[151] C.-L. Lui, T.-C. Fu, and T.-Y. Cheung, "Agent-based network intrusion detection system using data mining approaches," in *Information Technology and Applications, 2005. ICITA 2005. Third International Conference on*, vol. 1, july 2005, pp. 131 – 136 vol.1.

[152] M. Chirumamilla and B. Ramamurthy, "Agent based intrusion detection and response system for wireless lans," in *Communications, 2003. ICC '03. IEEE International Conference on*, vol. 1, may 2003, pp. 492 – 496 vol.1.

[153] D. Dasgupta, F. Gonzalez, K. Yallapu, J. Gomez, and R. Yarramsettii, "Cids: An agent-based intrusion detection system," *Computers & Security*, vol. 24, no. 5, pp. 387 – 398, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404805000179

[154] C. Krgel, "Sparta - a mobile agent based intrusion detection system," in *Proceedings of the IFIP conference on Network Security (I-NETSEC)*. Kluwer Academic Publishers, 2001.

[155] J. Balasubramaniyan, J. Garcia-Fernandez, D. Isacoff, E. Spafford, and D. Zamboni, "An architecture for intrusion detection using autonomous agents," in *Computer Security Applications Conference, 1998, Proceedings., 14th Annual*, dec 1998, pp. 13 –24.

[156] Y. Mo, Y. Ma, and L. Xu, "Design and implementation of intrusion detection based on mobile agents," in *IT in Medicine and Education, 2008. ITME 2008. IEEE International Symposium on*, dec. 2008, pp. 278 –281.

[157] J. Sen, I. Sengupta, and P. Chowdhury, "An architecture of a distributed intrusion detection system using cooperating agents," in *Computing Informatics, 2006. ICOCI '06. International Conference on*, june 2006, pp. 1 –6.

[158] J. M. Bradshaw, *An introduction to software agents*. Cambridge, MA, USA: MIT Press, 1997, pp. 3–46.

[159] OpenNebula. (2012) Contextualizing Virtual Machines 2.2. [Online]. Available: http://opennebula.org/documentation:archives:rel2.2:cong-Accessed:10.06.2013

[160] JADE Web Site. Java Agent Development Platform. [Online]. Available: http://jade.tilab.com-Accessed:01.10.2013

[161] Foundation for Intelligent Physical Agents. Agent communication language specification (acl). [Online]. Available: http://www.fipa.org/repository/aclspecs.html-Accessed:11.10.2013

[162] David Grimshaw. JADE Administration Tutorial. [Online]. Available: http://jade.tilab.com/doc/tutorials/JADEAdmin-Accessed:10.06.2013

[163] S. Staniford-chen, S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagl, K. Levitt, C. Wee, R. Yip, and D. Zerkle, "GrIDS - A Graph Based Intrusion Detection System For Large Networks," in *In Proceedings of the 19th National Information Systems Security Conference*, 1996, pp. 361–370.

[164] E. Cortese, F. Quarta, G. Vitaglione, T. I. Lab, C. Direzionale, J. Message, and T. System, "Scalability and performance of jade message transport system," 2002.

[165] BootChart. (2012). [Online]. Available: http://wiki.ubuntuusers.de/BootChart-Accessed:31.10.2013

[166] F. Doelitzscher, T. Ruebsamen, T. Karbe, M. Knahl, C. Reich, and N. Clarke, "Sun behind Clouds - On Automatic Cloud Security Audits and a Cloud Audit Policy Language," in *International Journal on Advances in Networks and Services*, T. Gyires, Ed., vol. 6, no. 12, 2013.

[167] Nagios. IT Infrastructure Monitoring. [Online]. Available: http://www.nagios.org-Accessed:24.06.2013

[168] Nagios Inc. Nagios Plugins. [Online]. Available: http://www.nagios.org/projects/nagiosplugins-Accessed:01.10.2013

[169] M. Al-Morsy and H. Faheem, "A new standard security policy language," *Potentials, IEEE*, vol. 28, no. 2, pp. 19 –26, march-april 2009.

[170] J. A. Hoagland, R. Pandey, R. P, and K. N. Levitt. A Graph-based Language for Specifying Security Policies. [Online]. Available: http://www.nr.no/~abie/Papers/LaSCO-DCCA-7-submit.pdf-Accessed:01.10.13

[171] L. Kagal. (2004) Rei Ontology Specifications. [Online]. Available: http://www.csee.umbc.edu/~lkagal1/rei/-Accessed:10.06.2013

[172] Policy Group - Department of Computing Imperial College London. Ponder: A policy language for distributed systems management. [Online]. Available: http://www-dse.doc.ic.ac.uk/Research/policies/ponder.shtml-Accessed06.09.2013

[173] ——. Ponder2. [Online]. Available: http://www.ponder2.net-Accessed:10.09.2013

[174] *OASIS eXtensible Access Control Markup Language (XACML) TC*, Organization for the Advancement of Structured Information Standards (OASIS) Std. [Online]. Available: http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf-Accessed:01.10.2013

[175] *SPARQL Protocol And RDF Query Language*, RDF Data Access Working Group (DAWG) Std.

[176] OVAL - Open Vulnerability and Assessment Language. [Online]. Available: https://oval.mitre.org/about/documents.html#language-Accessed:01.10.2013

[177] Organization for the Advancement of Structured Information Standards. (2012) Web Services Business Process Execution Language (WSBPEL) TC. [Online]. Available: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel-Accessed:31.10.2013

[178] Workflow and Agents Development Environment (WADE). [Online]. Available: http://jade.tilab.com/wade-Accessed01.10.203

[179] Internet Engineering Task Force. RFC 4765 - The Intrusion Detection Message Exchange Format (IDMEF). [Online]. Available: http://tools.ietf.org/html/rfc4765-Accessed:01.10.2013

[180] T. Karbe, "Design and Development of an Audit Policy Language for Cloud Computing Environments," Cloud Research Lab - University of Apllied Sciences Furtwangen, Tech. Rep., 2013. [Online]. Available: http://wolke.hs-furtwangen.de/publications/theses

[181] Distributed Management Taskforce Inc. (DMTF). (2013) Common information model (cim). [Online]. Available: http://dmtf.org/standards/cim-Accessed04.09.2013

[182] DMTF Policy Working Group. CIM Schema Final Documentation. [Online]. Available: http://dmtf.org/standards/cim/cim_schema_v2340-Accessed:10.06.2013

[183] Distributed Management Taskforce Inc. (DMTF). (2006) Distributed management task force (dmtf) tutorial. [Online]. Available: http://www.wbemsolutions.com/tutorials/DMTF/dmtftutorial.pdf-Accessed:04.09.2013

[184] MS TechNet. (2004, 09) Windows Management Instrumentation. [Online]. Available: http://www.microsoft.com/germany/technet/datenbank/articles/600682.mspx-Accessed:10.06.2013

[185] SBLIM. (2009) SBLIM Project Wiki. [Online]. Available: http://sourceforge.net/apps/mediawiki/sblim/index.php?title=MainPage-Accessed:10.06.2013

[186] Distributed Management Taskforce Inc. (DMTF). Cloud Infrastructure Management Interface (CIMI). Accessed: 10.06.2013. [Online]. Available: http://www.dmtf.org/standards/cloud-Accessed:10.06.2013

[187] Distributed Management Task Force Inc. Cloud management initiative. [Online]. Available: http://dmtf.org/standards/cloud-Accessed:07.09.2013

[188] Distributed Management Taskforce Inc. (DMTF). Cloud infrastructure management interface (cimi) primer. [Online]. Available: http://dmtf.org/sites/default/files/standards/documents/DSP2027_1.0.1.pdf-Accessed10.09.2013

[189] ——. Cimi xml schema. [Online]. Available: http://schemas.dmtf.org/cimi/1/dsp8009_1.0.1.xsd-Accessed:10.09.2013

[190] ——. (2007, 02) Policy Profile. [Online]. Available: http://www.dmtf.org/sites/default/files/standards/documents/DSP1003.pdf-Accessed:10.06.2013

[191] ——. Cim schema - policy model. [Online]. Available: http://www.wbemsolutions.com/tutorials/CIM/cim-model-policy.html-Accessed:10.06.2013

[192] F. Doelitzscher, M. Knahl, C. Reich, and N. Clarke, "Anomaly detection in iaas clouds," in *Inproceedings of the 5th IEEE International Conference on Cloud Computing Technology and Science*, 12 2013.

[193] P. Winter, H. Lampesberger, and M. Zeilinger, "Anomalieerkennung in Computernetzen," *Datenschutz und Datensicherheit-DuD*, pp. 235–239, 2011. [Online]. Available: http://www.springerlink.com/index/K12443031LQ84V55.pdf

[194] A. Banerjee, V. Chandola, V. Kumar, J. Srivastava, and A. Lazarevic, "Anomaly detection: A tutorial," in *SIAM International Conference on Data Mining*, Apr. 2008. [Online]. Available: http://www.siam.org/meetings/sdm08/TS2.ppt-Accessed:31.10.2013

[195] A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, and J. Srivastava, "A comparative study of anomaly detection schemes in network intrusion detection," in *Proceedings of the Third SIAM International Conference*

on Data Mining, vol. 3. SIAM, 2003, pp. 25–36. [Online]. Available: http://www.siam.org/proceedings/datamining/2003/dm03_03LazarevicA.pdf

[196] P. Garcia-Teodoro, J. Daaz-Verdejo, G. Macia-Fernandez, and E. Vazquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Computers Security*, vol. 28, no. 1-2, pp. 18–28, 2009. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0167404808000692

[197] A. Patcha and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Computer Networks*, vol. 51, no. 12, pp. 3448–3470, 2007. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S138912860700062X

[198] Massachusetts Intitute of Technology (MIT). Darpa intrusion detection data sets. [Online]. Available: http://www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/data-Accessed10.07.2013

[199] G. Nascimento and M. Correia, "Anomaly-based intrusion detection in software as a service," in *Dependable Systems and Networks Workshops (DSN-W), 2011 IEEE/IFIP 41st International Conference on*, Jun. 2011, pp. 19–24. [Online]. Available: http://dx.doi.org/10.1109/DSNW.2011.5958858

[200] L. Zhang, J. Wang, and S. Lin, "Design of the network traffic anomaly detection system in cloud computing environment," in *Information Science and Engineering (ISISE), 2012 International Symposium*, 2012.

[201] H. Pannu, J. Liu, and S. Fu, "Aad: Adaptive anomaly detection system for cloud computing infrastructures," in *Reliable Distributed Systems (SRDS), 2012 IEEE 31st Symposium on*, 2012, pp. 396–397.

[202] S. Fu, "Performance metric selection for autonomic anomaly detection on cloud computing systems," in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, 2011, pp. 1–5.

[203] A. Maithili, R. V. Kumari, and S. Rajamanickam, "Neural networks cum cloud computing approach in diagnosis of cancer," in *International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622*, vol. 2, no. 2, 2012, pp. 428–435.

[204] Z. Mahmood, C. Agrawal, S. S. Hasan, and S. Zenab, "Intrusion detection in cloud computing environment using neural network," in *International Journal of Research in Computer Engineering and Electronics*, vol. 1, no. 1, 2012, iSSN: 2319-376X.

[205] B. Joshi, A. Vijayan, and B. Joshi, "Securing cloud computing environment against ddos attacks," in *Computer Communication and Informatics (ICCCI), 2012 International Conference on*, 2012, pp. 1–5.

[206] "OWASP top 10 - 2010. The ten most critical web application security risks," OWASP The Open Web Application Security Project, Tech. Rep., 2010. [Online]. Available: http://owasptop10.googlecode.com/files/OWASP%20Top%2010%20-%202010.pdf-Accessed01.10.2013

[207] A. Dainotti, A. King, k. Claffy, F. Papale, and A. Pescapè, "Analysis of a "/0" stealth scan from a botnet," in *Proceedings of the 2012 ACM conference on Internet measurement conference*, ser. IMC '12. New York, NY, USA: ACM, 2012.

[208] S. A. Hofmeyr, S. Forrest, and A. Somayaji, "Intrusion detection using sequences of system calls," *Journal of Computer Security*, vol. 6, pp. 151–180, 1998.

[209] J. Cannady, "Artificial neural networks for misuse detection," in *NATIONAL INFORMATION SYSTEMS SECURITY CONFERENCE*, 1998, pp. 443–456.

[210] Amazon Web Services. Regions and Availability Zones. [Online]. Available: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html-Accessed11.10.2013

[211] Christopher M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1996.

[212] Y. Xiang, W. Zhou, and M. Guo, "Flexible deterministic packet marking: An ip traceback system to find the real source of attacks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 20, no. 4, pp. 567–580, 2009.

[213] Thomas Jetter. neural network editor membrain. [Online]. Available: http://www.membrain-nn.de-Accessed:19.06.2013

[214] P. Frasconi, M. Gori, and A. Tesi, "Successes and failures of backpropagation: A theoretical investigation," in *Progress in Neural Networks. Ablex Publishing*. Ablex Publishing, pp. 205–242.

[215] J. P. Planquart, "Application of neural networks to intrusion detection," SANS Institute, Tech. Rep., 2001.

[216] HP systems insight manager. [Online]. Available: http://www.hp.com/go/sim-Accessed:19.06.2013

[217] IBM. Tivoli software. [Online]. Available: http://www.ibm.com/tivoli-Accessed:02.07.2013

[218] OpenNebula. Ganglia Monitoring 4.0. [Online]. Available: http://opennebula.org/documentation:rel4.0:ganglia-Accessed:24.06.2013

[219] C. Wang, V. Talwar, K. Schwan, and P. Ranganathan, "Online detection of utility cloud anomalies using metric distributions," in *Network Operations and Management Symposium (NOMS), 2010 IEEE*, 2010.

[220] S. Pervez, I. Ahmad, A. Akram, and S. U. Swati, "A comparative analysis of artificial neural network technologies in intrusion detection systems," in *Proceedings of the 6th WSEAS International Conference on Multimedia, Internet & Video Technologies*, ser. MIV'06. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2006, pp. 84–89. [Online]. Available: http://dl.acm.org/citation.cfm?id=1365598.1365612

[221] O. Agmon Ben-Yehuda, M. Ben-Yehuda, A. Schuster, and D. Tsafrir, "Deconstructing amazon ec2 spot instance pricing," in *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*, 2011, pp. 304–311.

[222] A. Monfared and M. Jaatun, "Monitoring intrusions and security breaches in highly distributed cloud environments," in *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*, 2011, pp. 772–777.

[223] S. Phillips, V. Engen, and J. Papay, "Snow white clouds and the seven dwarfs," in *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*, 2011, pp. 738–745.

[224] J. Lecomte, N. Clarke, and S. Furnell, "Artificial impostor profiling for keystroke dynamics on a mobile handset," in *Proceedings of Fifth International Network Conference (INC 2005)*, 07 2005, pp. 199–206.

[225] Cloud Research Lab Furtwangen University. Security Audit as a Service. [Online]. Available: http://wolke.hs-furtwangen.de/currentprojects/saaas-Accessed:01.10.2013

[226] ——. StudiCloud at HFU. [Online]. Available: http://www.wolke.hs-furtwangen.de/currentprojects/studi-cloud-Accessed:01.10.2013

[227] ——. Accountability for Cloud and Other Future Internet Services. [Online]. Available: http://wolke.hs-furtwangen.de/currentprojects/a4cloud-Accessed:01.10.2013

[228] ——. Autonomic SLA Management as a Service for Cloud Services. [Online]. Available: http://www.wolke.hs-furtwangen.de/currentprojects/aslamaas-Accessed:01.10.2013

[229] ——. Ambient Assisted Living: Person Centred Environment for Information, Communication and Learning. [Online]. Available: http://www.wolke.hs-furtwangen.de/currentprojects/pceicl-Accessed:01.10.2013

[230] Haydn Solomon. How you can use qemu/kvm base images to be more productive (Part 1). [Online]. Available: http://www.linux-kvm.com/content/how-you-can-use-qemukvm-base-images-be-more-productive-part-1-Accessed11.10.2013

[231] The Debian Project. Debian Linux. [Online]. Available: http://www.debian.org-Accessed:11.10.2013

[232] Google Inc. The Google Web Toolkit. [Online]. Available: http://code.google.com/p/google-web-toolkit-Accessed:11.10.2013

[233] Apache Software Foundation. Apache Tomcat. [Online]. Available: http://tomcat.apache.org-Accessed:11.10.2013

[234] Cloud Research Lab Furtwangen University. StudiCloud Source Forge project. [Online]. Available: http://sourceforge.net/projects/studicloud/-Accessed:01.10.2013

[235] IEEE SERVICES 2012 8th IEEE World Congress on Services. Winners of Service Cup 2012. [Online]. Available: http://users.encs.concordia.ca/~yuhong/ServiceCup2012/Results.html-Accessed:19.09.2013

[236] Roy Thomas Fielding, "Architectural Styles and the Design of Network-based Software Architectures," Ph.D. dissertation, University of California, Irvine, 2000. [Online]. Available: http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm-Accessed:01.10.2013

[237] T. Berners-Lee, R. Fielding, and L. Masinter, "Uniform resource identifier (uri): Generic syntax," RFC 3986 (INTERNET STANDARD), Internet Engineering Task Force, Jan. 2005. [Online]. Available: http://www.ietf.org/rfc/rfc3986.txt-Accessed:31.10.2013

[238] inotify. - monitoring file system events. [Online]. Available: http://linux.die.net/man/7/inotify-Accessed:31.10.2013

[239] The Availability Digest, "Amazon's Availability Zones," Tech. Rep., 2011. [Online]. Available: http://www.availabilitydigest.com/public_articles/0611/amazon_avilability_zones.pdf-Accessed01.10.2013

[240] BBC News. (2013) NSA: New reports in German media deepen US-Merkel spy row. [Online]. Available: http://www.bbc.co.uk/news/world-europe-24692908-Accessed:07.11.2013

[241] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext transfer protocol – http/1.1," RFC 2616 (Draft Standard), Internet Engineering Task Force, Jun. 1999, updated by RFCs 2817, 5785, 6266, 6585. [Online]. Available: http://www.ietf.org/rfc/rfc2616.txt

Glossary

A6 Is a project to provide an interface and namespace for automated Audit, Assertion, Assessment, and Assurance of cloud infrastructures

ACL Agent Communication Language

AID Agent Identifier

AMM Agent Mobility Manager

AMS Agent Management System

API Application programming interface, a source code-based specification which can be used as an interface by software components to communicate with each other

AWS Amazon Web Services

CAPTCHA Completely Automated Public Turing test to tell Computers and Humans Apart. A test, commonly used on web sites to check if the user is human

CloudIA Cloud Infrastructure and Applications - a Cloud infrastructure operated at Furtwangen University

Cloud customer See Cloud user

Cloud user A Cloud user is a customer of a Cloud provider utilising Cloud resource, such as virtual machines

CSP A cloud service provider is a provider of cloud services

CSA The Cloud Security Alliance is a non-profit organisation that promotes research into best practices for securing cloud computing

DF Directory Facility

CMS Cloud Management System

DoS Denial of Service, an attempt to make a computer or network resource unavailable through a huge amount of legitimated requests

DDoS Distributed Denial of Service, an attempt to make a computer or network resource unavailable through a huge amount of legitimated requests from distributed sources

EC2 Elastic Computing Cloud - IaaS Cloud infrastructure of Amazon Web Services

FIPA Foundation for Intelligent Physical Agents

GUI Graphical User Interface

HFU Hochschule Furtwangen University

HTTP Hyper Text Transfer Protocol

JVM Java Virtual Machine

KVM Kernel-based Virtual Machine, a full virtualization solution for Linux supporting CPU virtualization extensions (Intel VT or AMD-V)

IaaS Infrastructure as a Service, a model where a virtual IT infrastructure is rented by a user from a provider as a service

ICT Information and communications technology, or information and communication technology

ISO 27001 Is an Information Security Management System standard published by the International Organisation for Standardisation (ISO) and the International

ITIL Information Security Management System, a set of policies concerned with information security management as defined by ISO/IEC 27001 standards Electrotechnical Commission (IEC)

JADE Java Agent Development Framework

MTP Message Transfer Protocol

OS Operating System

PaaS Platform as a Service

RDP Remote Desktop Protocol, a proprietary protocol developed by Microsoft, providing user with a graphical interface to other computers

RPC Remote Procedure Call

SAaaS Security Audit as a Service, an infrastructure to support IT security audits of cloud computing infrastructures

SaaS Software as a Service, a model of software deployment where users rent an application from a provider and use it as a service

SSH Secure Shell, a network communication protocol for secure data communication

TLS Transport Layer Security

VM Virtual Machine

VMware Is a full virtualization solution for Linux, Windows and Mac OS

Xen Is a hardware- and para virtualization solution for Linux

XML eXtensible Markup Language

Appendix

The Appendix provides additional information and details on the presented topics of the research. It is organised with the following structure:

Appendix A1 - Content of attached DVD-ROM

Appendix A2 - Overview over all developed SAaaS Agents

Appendix A3 - Specification of Cloud Audit Policy Language (CAPL)
Part 4.1 - CAPL Class Diagram
Part 4.2 - CAPL Class Description
Part 4.2 - CAPL Server Return Values

Appendix A4 - Bootchart Analysis of SAaaS Agents Platform Overhead

Appendix A5 - Email Correspondence with Public Cloud Provider

Appendix A6 - Additional Material on Cloud Usage Simulator

Appendix A7 - Publication - Conference Poster

A.1 DVD ROM Content

The attached DVD contains the following material within the corresponding folders (listed alphabetically):

- `CloudAuditPolicyLanguage/`
  - Cloud Audit Policy Language Java Doc
- `CloudUsageSimulator/`
  - Simulator Source code
  - Simulator executable
  - Simulation results
- `Paper/`
  - Full text of published paper
- `Phd Thesis/`
  - Security Audit Compliance for Cloud Computing - PDF version
  - Security Audit Compliance for Cloud Computing - Txt version
  - LaTeX source code of thesis document
- `Poster/`
  - IEEE SERVICES 2012 conference poster
- `SAaaS_Prototype_Demos/`
  - Video of SAaaS Demonstrations
- `Videos/`
  - Video of Simulation

A.2 Developed SAaaS Agents

The following provides an overview over all developed SAaaS agents.

InotifyAgent

This agent monitors file system changes and sends events to Event Aggregator agent.

Name	inotifyAgent	
Position	Target VM / cloud host	
Source code	./trunk/jade/src/de/hfu/wolke/saaas/demo1/InotifyAgent.java	
Scenario	SAaaS Demo 1, SAaaS Demo 2	
Requirements on target	Inotify	Linux kernel with inotify support
Requirements on JADE platform	configs/InotifyWatchService.properties configs/ConfigWatchService.properties bin-32/libinotify-java.so bin-64/libinotify-java.so	list off files to monitor list off files to monitor 32bit library for kernel communication 64bit library for kernel communication

SocketProxyAgent

This agent receives ACL messages from a socket and forwards them to its receiver, which needs to be listed in spa.conf.

Name	SocketProxyAgent	
Position	Target VM / cloud host	
Source code	./trunk/jade/src/jade/tools/SocketProxyAgent/SocketProxyAgent.java	
Used by	LoginBruteForceDetectionAgent, ManagementConnectionAgent, CMSActivityAgent	
Requirements on target	spa.inf /etc/default/jade	Contains agents which are allowed to receive messages from SocketProxyAgent Agent gets started with platform
Requirements on JADE platform		

SAaaSGateWayAgent

This agent enables communication between non-jade applications and jade agents. Messages in ACL format can be send to a jade platform through this agent.

Name	SAaaSGateWayAgent	
Position	Cloud GUI	
Source code	./trunk/jade/src/de/hfu/wolke/saaas/server/SAaaSGateWayAgent.java	
Used by	all	
Requirements on target		
Requirements on JADE platform	saaasdb	Table saaasglobals in saaasdb

ManagerAgent

This agent manages agent communication. Events get saved into event database and displayed in SAaaS GUI.

Name	ManagerAgent	
Position	SAaaS-Agent Management Platform (global & user specific)	
Source code	./trunk/jade/src/de/hfu/wolke/saaas/demo1/ManagerAgent.java	
Sccenario	Event Management	
Requirements on target		
Requirements on JADE platform	jade1.sh saaasdb	Agent gets started with platform Table saaasglobals in saaasdb

VMMonitoringAgent

This agent monitores the Cloud Management System's log file and forwards events to other agents.

Name	VMMonitoringAgent	
Position	Cloud Management System	
Source code	./trunk/jade/src/de/hfu/wolke/saaas/vmmonitoring/VMMonitoringAgent.java	
Scenario	Monitoring of Open Nebula log file	
Requirements on target	configs/ONELogFiles.properties	Contains path to OpenNebula log file
Requirements on JADE platform	saaasdb	Table saaasglobals and vmmonitoring in saaasdb

ConfigAuditAgent

This agent gets deployed on-demand by Managemer Agent. It contains a task list, which gets worked through. After the list is done, it deletes itself.

Name	ConfigAuditAgent	
Position	Target VM / cloud host	
Source code	./trunk/jade/src/de/hfu/wolke/saaas/demo1a/ConfigAuditAgent.java	
Scenario	Demo1a and SAaaS Demo 2	
Requirements on target		
Requirements on JADE platform	saaasdb	Table saaasglobals in saaasdb

ClamavAgent

This agent gets deployed on-demand by Managemer Agent. It contains a file list, which gets transmitted to virus scanner clamav. After the scan is done, it deletes itself.

Name	ClamavAgent	
Position	Target VM / cloud host	
Source code	./trunk/jade/src/de/hfu/wolke/saaas/demo1a/ClamAVAgent.java	
Scenario	Demo1a and SAaaS Demo 2	
Requirements on target	ClamAV	ClamAV needs to be installed and listening on port TCP 3310
Requirements on JADE platform	saaasdb	Table saaasglobals in saaasdb

WeblogAgent

This agent monitors log file of apache web server. In case a not listed file gets accessed, an event is send.

Name	WeblogAgent	
Position	Target VM / cloud host	
Source code	./trunk/jade/src/de/hfu/wolke/saaas/demo1/WeblogAgent.java	
Scenario	SAaaS Demo1	
Requirements on target	Apache web server	
Requirements on JADE platform	saaasdb configs/LogFiles.properties	Table saaasglobals in saaasdb Contains path to Apache log file

RolloutManagerAgent

This agent is responsible for moving agents to other agent platforms.

Name	RolloutManagerAgent	
Position	SAaaS-Agent Management Platform (global & user specific)	
Source code	./trunk/jade/src/de/hfu/wolke/saaas/demo1/AgentRolloutManager.java	
Scenario	All	
Requirements on target		
Requirements on JADE platform		

CMSActivityAgent

This agent monitors the Cloud Management System's logfile and formwars events to the AnomalyDetectionAgent.

Name	CMSActivityAgent	
Position	Cloud Management System	
Source code	./trunk/jade/src/de/hfu/wolke/saaas/shared/CMSActivityAgent.java	
Scenario	SAaaS Demo3b	
Requirements on target		
Requirements on JADE platform	saaasdb spa.inf	Table saaasglobals in saaasdb

EventAggregatorAgent

This agent collects all sensor agent events and pre-processes them before forwarding. In SAaaS Demo 1 the agent forwards only messages in case no authorized user is still logged on to target VM.

Name	EventAggregatorAgent	
Position	Target VM / cloud host	
Source code	./trunk/jade/src/de/hfu/wolke/saaas/demo1/EventAggregatorAgent.java	
Scenario	SAaaS Demo1	
Requirements on target		
Requirements on JADE platform	saaasdb ManagementConnectionAgent	Table saaasglobals in saaasdb To check if user is logged in

ManagementConnectionAgent

This agent monitors if a user is still logged in on a target VM.

Name	ManagementConnectionAgent	
Position	Target VM / cloud host	
Source code	./trunk/jade/src/de/hfu/wolke/saaas/demo2/ManagementConnectionAgent.java	
Scenario	SAaaS Demo2	
Requirements on target	/var/run/utmp /etc/pam.d/common-session /usr/local/bin/notify-login spa.inf SocketProxyAgent	Detection of logged-in users Gets modified by agent to support monitoring features Gets executed by PAM script to send event to SocketProxyAgent
Requirements on JADE platform	saaasdb	Table saaasglobals in saaasdb

LoginBruteForceDetectionAgent

This agent sends events about user login attempts.

Name	LoginBruteForceDetectionAgent	
Position	SAaaS-Agent Management Platform (global & user specific)	
Source code	./trunk/jade/src/de/hfu/wolke/saaas/demo2/LoginBruteForceDetectionAgent.java	
Scenario	SAaaS Demo3	
Requirements on target	/etc/pam.d/common-auth /usr/bin/pam-login spa.inf SocketProxyAgent	Gets modified by agent to support monitoring features Gets executed by PAM script to send event to SocketProxyAgent
Requirements on JADE platform	saaasdb jade1.sh	Table saaasglobals in saaasdb Agent gets started with platform

ScalabilityEventAgent

This agent gets started in case a scalability event gets detected by the CMSActivityAgent. It the creates MetricAgents for checking scalability metrics.

Name	ScalabilityEventAgent	
Position	SAaaS-Agent Management Platform (global & user specific)	
Source code	./trunk/jade/src/de/hfu/wolke/saaas/scaling/ScalabilityEventAgent.java	
Scenario	SAaaS Demo2	
Requirements on target		
Requirements on JADE platform	saaasdb	Table saaasglobals in saaasdb

MetricAgents

This agent receives a metric to be checked.

Name	MetricAgents	
Position	Target VM / cloud host	
Source code	./trunk/jade/src/de/hfu/wolke/saaas/scaling/MetricAgent.java	
Scenario	SAaaS Demo2	
Requirements on target		
Requirements on JADE platform	ScalabilityEventAgent	Starts metric agents, events get reported back

CPUAgent
This agent is used during the SAaaS performance tests to measure CPU usage.

Name	CPUAgent
Position	Target VM / cloud host
Source code	./trunk/jade/src/de/hfu/wolke/saaas/performance/CPUAgent.java
Scenario	Jade performance tests

MoveAgent
This agent is used during the SAaaS performance tests to measure agent deploy time.

Name	MoveAgent
Position	Target VM / cloud host
Source code	./trunk/jade/src/de/hfu/wolke/saaas/performance/MoveAgent.java
Scenario	Jade performance tests

ReceiverAgent
This agent is used during the SAaaS performance tests to measure agent deploy time.

Name	ReceiverAgent
Position	Target VM / cloud host
Source code	./trunk/jade/src/de/hfu/wolke/saaas/performance/ReceiverAgent.java
Scenario	Jade performance tests

SenderAgent
This agent is used during the SAaaS performance tests to measure agent deploy time.

Name	SenderAgent
Position	Target VM / cloud host
Source code	./trunk/jade/src/de/hfu/wolke/saaas/performance/SenderAgent.java
Scenario	Jade performance tests

A.3 Cloud Audit Policy Language Specification

This is a comprehensive description of all developed classes of the Cloud Audit Policy Language CAPL. An electronic Javadoc documentation can be found on the CD attached to this PhD thesis. For completeness, the CAPL class diagram, introduced in Chapter 5 is presented again.

CAPL Class Diagram

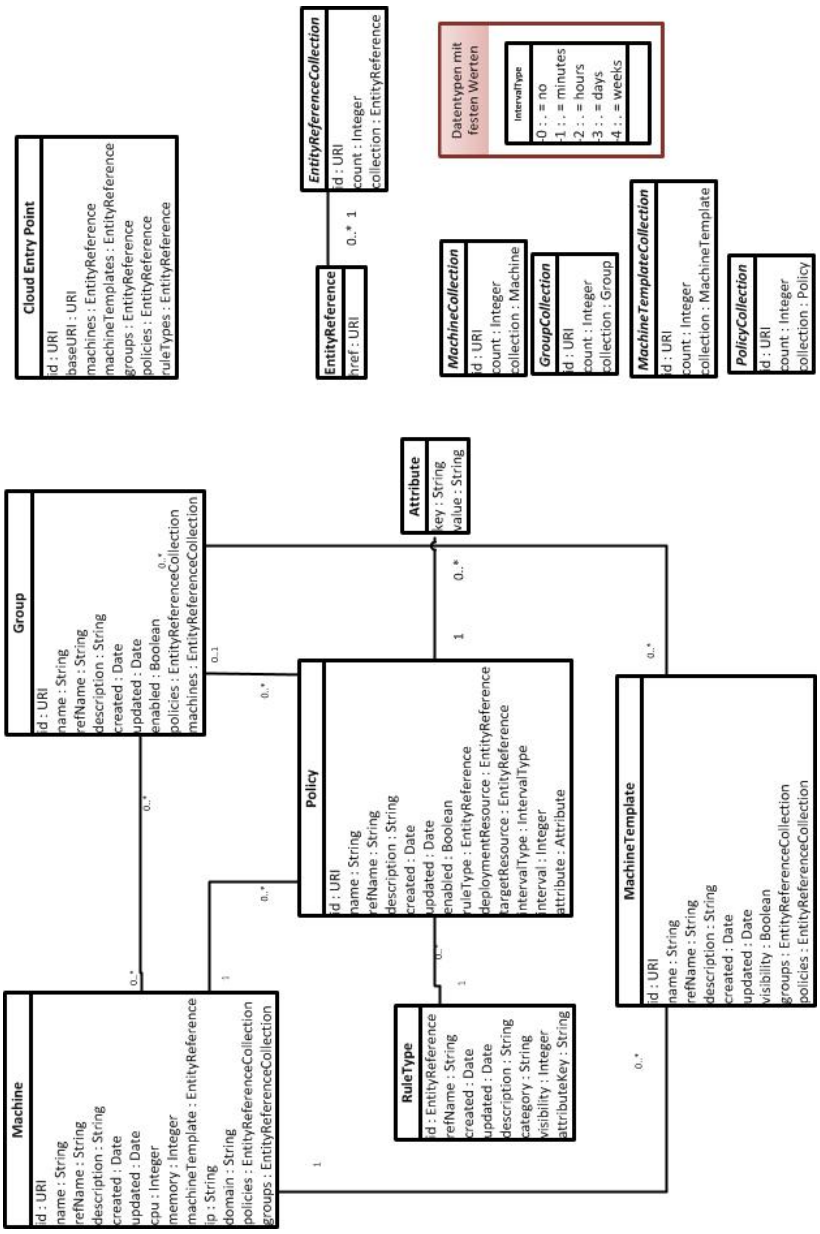


Figure A.1: CAPL class diagram

CAPL Class Description

The following classes are described:

- CloudEntryPoint
- AbstractCollection
- Machine
- MachineCollection
- MachineTemplate
- Group
- GroupCollection
- RuleType
- RuleTypeCollection
- Policies
- PolicySet
- PolicyCollection
- Action
- ActionCollection

Basic Class Structure

Each CAPL class consists of the following basic structure of attributes. They have the same meaning in every class, thus they are going to be presented here once and not included every time in the description of the corresponding classes. Table A.1 provides a definition of the basic class attributes.

Data Type	Name	Description
resourceURI	URI	Contains the URI for identification of type within XML schema
Id	URI	Object ID under which it can be referenced
name	String	Name of instance
description	String	Description of corresponding instance
created	Date	Gets automatically set at object creation time
updated	Date	Gets automatically updated at object modification time

Table A.1: CAPL classes: basic attribute structure

Cloud Entry Point

Forms an interface for accessing the CAPL server. It provides an overview over all available resources and their corresponding URIs for a CAPL client (SAaaS Policy Modeller). It does not use the same basic structure, described in Table A.1. For the SAaaS prototype (see Chapter 7) it is accessible at `https://cloud.hs-furtwangen.de/CAPLPrototyp/rest/`. Table A.2 describes the attributes of the Cloud Entry Point.

AbstractCollection

For every class an additional class with the name schema `[Class-Name]Collection` is implemented, for providing lists and instances of this very own class. These "collection-classes" inherit from an abstract class `AbstractCollection`. Thus, each single class has the same structure as class AbstractCollection. No additional attributes exist. In case, a class contains a collection as an attribute, it is possible to get a list of instances

Data Type	Name	Description
id	String	ID of Cloud Entry Point, e.g. `https://research.cloud.hs-furtwangen.de/ CAPLPrototyp/rest/CEP`
baseURI	URI	Contains the base URI, e.g. `https://research.cloud.hs-furtwangen.de/CAPLPrototyp/rest/`
machines	EntityReference	A reference for MachineCollection, which delivers all MachineTemplate instances, e.g. `https://research.cloud.hs-furtwangen.de/CAPLPrototyp/rest/MachineTemplates`
groups	EntityReference	A reference for GroupCollection, which delivers all group instances, e.g. `https://research.cloud.hs-furtwangen.de/CAPLPrototyp/rest/groups`
policies	EntityReference	A reference for PolicyCollection, which delivers all policy instances, e.g. `https://research.cloud.hs-furtwangen.de/CAPLPrototyp/rest/policies`
ruleTypes	EntityReference	A reference to the list of all available RuleType, e.g. `https://research.cloud.hs-furtwangen.de/CAPLPrototyp/rest/ruleTypes`

Table A.2: CAPL class: Cloud Entry Point

or references (`EntityReferenceCollection`) for this specific class. Thus, only data which is necessary gets queried and submitted, saving unnecessary data overhead. Table A.3 provides a definition of AbstractCollection.

Data Type	Name	Description
id	URI	ID of Collection under which it can be referenced
count	Integer	Number of objects in this list
collection	List	A list of instances of this data type. Example: In case of `MachineCollection` it includes multiple instances of type `Machine`

Table A.3: CAPL class: AbstractCollection

Machine

A `Machine` is an instance which needs to be audited. This can be a physical Cloud host or a virtual machine. Audit `Policies` can be assigned to a `Machine`. `Machines` can be logically grouped together via the class `Group`. Machines are created by reading information from the Cloud management system which hosts and VMs exist. A definition of Machine is provided in Table A.4.

Data Type	Name	Description
resourceURI	URI	Contains the base URI, e.g. `https://research.cloud.hs-furtwangen.de/CAPLPrototyp/rest/machines`
cpu	Integer	Number of CPUs
memory	Integer	Size of RAM
machineTemplate	EntityReference	A reference to MachineTemplate, which this instance is based on
ip	String	IP of instance
domain	String	Domain of instance
policies	EntityReferenceCollection	A list with references to policies which apply for this instance
groups	EntityReferenceCollection	A list of references to groups this instance is a member of

Table A.4: CAPL class: Machine

MachineCollection

Delivers a list of instances of type `Machine`. Table A.5 provides a definition of all attributes.

Data Type	Name	Description
id	URI	ID of Collection under which it can be referenced
count	Integer	Number of objects in this list
collection	Machine	A list of `Machine` instances

Table A.5: CAPL class: MachineCollection

MachineTemplate

MachineTemplate was adopted from CIMI but altered for the SAaaS scenario. Since VMs get configured by the Cloud management system, the CIMI attribute `MachineConfiguration` was not used. The class describes, on which template a `Machine` is based on and provides information on the operating system of this `Machine`. A Definition is provided in Table A.6.

Data Type	Name	Description
resourceURI	URI	Contains the base URI, e.g. `https://research.cloud.hs-furtwangen.de/CAPLPrototyp/rest/machineTemplates`
visibility	Boolean	Defines if this MachineTemplate is available for all Cloud user or only for the Cloud provider
groups	EntityReference	A list of references on available groups. Machines, based on this template are automatically assigned to these groups
policies	EntityReference	A list of references on policies. They get duplicated during creation time of a VM and attached to this `Machine` Therefore, policies for new VMs can be defined

Table A.6: CAPL class: MachineTemplate

MachineTemplateCollection

Delivers a list of instances of type `MachineTemplate`. Table A.7 provides a definition of all attributes.

Data Type	Name	Description
id	URI	ID of Collection under which it can be referenced
count	Integer	Number of objects in this list
collection	Machine	A list of `MachineTemplate` instances

Table A.7: CAPL class: MachineTemplateCollection

Group

The class `Group` provides the posibility to group `Policies` or `Machines`. All assigned policies of a certain `Group` apply for all `Machines` of this group.
Table A.8 provides attributes of class `Group`.

Data Type	Name	Description
resourceURI	URI	Contains the base URI, e.g. `https://research.cloud.hs-furtwangen.de/CAPLPrototyp/rest/groups`
enabled	Boolean	Defines if group is enabled or not (not implemented in SAaaS prototype)
policies	EntityReference	A list of references on policies which are assigned to `Machines` of this group
machines	EntityReference	A list of `Machines`, which are member of this group. All member inherit all policies of this group

Table A.8: CAPL class: Group

GroupCollection

Delivers a list of instances of type `Group`. Table A.9 provides a definition of all attributes.

Data Type	Name	Description
id	URI	ID of Collection under which it can be referenced
count	Integer	Number of objects in this list
collection	Machine	A list of `Group` instances

Table A.9: CAPL class: GroupCollection

RuleType

`RuleType` defines the type of a rule represents the SAaaS agent type. For each existing agent type a RuleType exists. RuleTypes are managed by the Cloud provider. The class was defined to be very general, to support a variety of different rules. Since different rules contain different attributes, rulespecific attributes get defined by the the attribute `properties`. Properties need to be provided at creation of a policy. They are used as a context-based attribute. They are stored in a hash map and assigned to the target agent upon agent configuration time. They are representing an agent's configuration. Context-based attributes can be assigned to simple data types, thus the CAPL server can evaluate if a context-based attribute is provided correctly and can validate it. As an example a reference on a gropu can be assigned to the data type `GroupReference`. To provide information on available RuleTypes, a policy developer can access them from the CAPL server via a `GET` request. A definition of RuleTypes is given in Table A.10. For the SAaaS prototype, RuleTypes for the defined policy scenarios presented in Section 5.3 are implemented.

Data Type	Name	Description
resourceURI	URI	Contains the base URI, e.g. `https://research.cloud.hs-furtwangen.de/CAPLPrototyp/rest/ruleTypes`
category	String	Enables categorisation of RuleTypes for sorting possibilities
visibility	Boolean	Defines if a RuleType is accessible for Cloud user. Used for development, for beta RuleTypes during development
attributeKey	String	Context-based value, which needs to be defined in a policy.

Table A.10: CAPL class: RuleTypes

RuleTypeCollection

Delivers a list of instances of type `Group`. Table A.11 provides a definition of all attributes.

Data Type	Name	Description
id	URI	ID of Collection under which it can be referenced
count	Integer	Number of objects in this list
collection	Machine	A list of `RuleType` instances

Table A.11: CAPL class: RuleTypeCollection

Policies

Each policy contains the same basic structure, shown in Table A.12. Additionally, each policy contains policy specific attributes with context-based attributes. They are depending on the type of policy and defined by the attribute `RuleType`. They are resulting in an agent's configuration. `Policies` can be assigned to `Machines`, `MachineTemplates`, and `Groups`.

Data Type	Name	Description
resourceURI	URI	Contains the base URI, e.g. `https://research.cloud.hs-furtwangen.de/CAPLPrototyp/rest/policies`
enabled	Boolean	Defines if policy is enabled or not
deploymentRessource	EntityReference	Is used in case a policy gets executed by a different `Machine`.
targetRessource	EntityReference	`Machine`, `MachineTemplate` or `Group` a policy is assigned to
intervalType	IntervalType	Defines interval a policy should be executed
interval	Integer	Defines value of interval
attribute	Attribute	Key/value pair containing context-based attributes of policy. Context is defined by `RuleType`. Can be used multiple times within a policy.

Table A.12: CAPL class: Policies

PolicyCollection

Delivers a list of instances of type `Policy`. Table A.13 provides a definition of all attributes.

Data Type	Name	Description
id	URI	ID of Collection under which it can be referenced
count	Integer	Number of objects in this list
collection	Machine	A list of `Policy` instances

Table A.13: CAPL class: PolicyCollection

PolicySet

A `PolicySet` is a `Policy` which contains multiple policies. A PolicySet can be assigned to a `Machine`, a **MachineTemplate** and a `Group`. The attribute ConditionType defines, how the policies are connected with each other. Possible values are: D - Disjunction, or C - Conjunction. In case of a conjunct connection, all policies need to be fulfilled for the PolicySet to be fulfilled. This corresponds to a traditional `AND` connection. In case of a disjunctive connection, only on policy needds to be fulfilled fort he PolicySet to be fulfilled. A description of class PoliySet is provided in Table A.14

Data Type	Name	Description
resourceURI	URI	Contains the base URI
enabled	Boolean	Defines if PolicySet is enabled or not
conditionType	ConditionType	Defines Type of Condition. For SAaaS prototype values `D` or `C` are available
targetRessource	EntityReference	`Machine`, `MachineTemplate` or `Group` a policy is assigned to
policies	EntityReferenceCollection	List of URIs to PolicyRule or MetricRule
intervalType	IntervalType	Defines interval a PolicySet should be executed
interval	Integer	Defines value of interval

Table A.14: CAPL class: PolicySet

Action

`Action` enables the executed of operations. It is used for execution policies and if a `Machine` is added to a `Group`. Its definition is provided in Table A.15.

Data Type	Name	Description
resourceURI	URI	Contains the base URI
action	String	Defines type of action. Possible values for SAaaS prototype: • **addMachine** - add a `Machine` to a `Group` • **removeMachine** - removes a `Machine` from a `Group` • **run** - start audit
target	Reference	Contains resource which the action applies to. In case of `addMachine` it contains the `Machine` which to add to a `Group`.

Table A.15: CAPL class: Action

CAPL Server - Return Values

The following Table lists HTTP status codes, the CAPL server uses to communicate wit a CAPL client. They are oriented on RFC2616 [241]

Return Value	Description
200 - OK	Request was executed successfully, requested information were returned (GET)
201 - CREATED	Request was executed successfully, new resource was created and corresponding URI returned (POST)
202 - ACCEPTED	Request was accepted, but was not completely executed. A possible reason could be that execution needs more time, e.g. in case of a policy. It is used for execution of policies (POST)
204 - NO CONTENT	Request was executed successfully. No return information exist. Used in case of a DELETE or ADDMACHINE operation. (POST,DELETE)
400 - BAD REQUEST	Request was not understood by server
404 - NOT FOUND	Requested URI was not found
405 - NOT ALLOWED	Action not allowed for this URI
415 - UNSUPPORTED TYPE	The supplied type is not supported
500 - INTERNAL SERVER ERROR	An internal server error occured
501 - NOT IMPLEMENTED	Requested method not implemented

Table A.16: CAPL server: Implemented HTTP return values

CAPL Server POST - Resource creation

Line 1 - 11 show the client request, line 14 - 34 the server's answer. Line 6 - 11 are the payload of the request to create a new resource. The server delivers the newly created object back (line 24 - 34) and confirms the success full object creation with return value 201. The new object can now be referenced by the URI `https://research.cloud.hs-furtwangen.de /CAPLPrototyp/res/groups/46`.

```
1 <!--
2 Request-URL = https://research.cloud.hs-furtwangen.de/CAPLPrototyp/rest/groups
3 HTTP-Methode = POST
4 Header = user: mumann Content-Type: text/xml
5 -->
6 <Group xmlns="http://research.cloud.hs-furtwangen.de">
7   <name>WWW Cluster for Loadbalancer 1</name>
8   <refName>WWW-Cluster1</refName>
9   <description>Group of web servers </description>
10  <enabled>true</enabled>
11 </Group>
12
13
14 <!--
15 Response-Status = 201 Created
16 Header = ( Date: Sat, 01 Jun 2013 13:25:21 GMT
17 Server: Apache/2.2.16 (Debian)
18 Connection: Keep-Alive
19 Keep-Alive: timeout=15, max=100
20 Content-Length: 504
21 Content-Type: application/xml )
22 -->
23 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
24 <Group xmlns="http://research.cloud.hs-furtwangen.de">
25   <id>https://research.cloud.hs-furtwangen.de/CAPLPrototyp/rest/groups/46</id>
26   <name>WWW Cluster for Loadbalancer 1</name>
27   <refName>WWW-Cluster1</refName>
28   <description>
29    Cluster for Load Balancer 1
30   </description>
31   <created>2013-06-01T00:00:00+02:00</created>
32   <updated>2013-06-01T00:00:00+02:00</updated>
33   <enabled>true</enabled>
34 </Group>
```

Listing A.1: CAPL example: Create Group resource

CAPL Server POST - Execution of operations

A machine with `id 5` gets added to group 46. The server's reply is shown in line 9 - 15, telling that the request was executed successfully and no return information exist.

```
1 <!--Request-URL = https://research.cloud.hs-furtwangen.de/CAPLPrototyp/rest/
      groups/46
2 HTTP-Methode = POST
3 Header = user: doelitz Content-Type: text/xml -->
4 <Action xmlns="http://research.cloud.hs-furtwangen.de">
5   <action>addMachine</action>
6   <target href="http://research.cloud.hs-furtwangen.de/CAPLPrototyp/rest/
      machines/5" />
7 </Action>
8
9 <!-- Response-Status = 204 No Content
10 Header = ( Date: Sat, 01 Jun 2013 13:25:21 GMT
```

```
11 Server: Apache/2.2.16 (Debian)
12 Connection: Keep-Alive
13 Keep-Alive: timeout=15, max=100
14 Content-Length: 0
15 Content-Type: text/plain ) -->
```

Listing A.2: CAPL example: Add Machine id

DELETE - Deletion of resources

A resource send to this method gets deleted by the server. It must be definitely referenced by its id. Listing A.3 shows a delete request of object 46.

```
1  <!--
2 Request-URL = https://research.cloud.hs-furtwangen.de/CAPLPrototyp/rest/groups
      /46
3 HTTP-Methode = DELETE
4 Header = user: mumann
5 -->
6
7 <!--
8 Response-Status = 204 No Content
9 Header = ( Date: Sat, 01 Jun 2013 13:25:21 GMT
10 Server: Apache/2.2.16 (Debian)
11 Connection: Keep-Alive
12 Keep-Alive: timeout=15, max=100
13 Content-Length: 0
14 Content-Type: plain/text )
15 -->
```

Listing A.3: CAPL example: Delete Group resource

SAaaS Demo 2 . Communication in Upscale Scenario

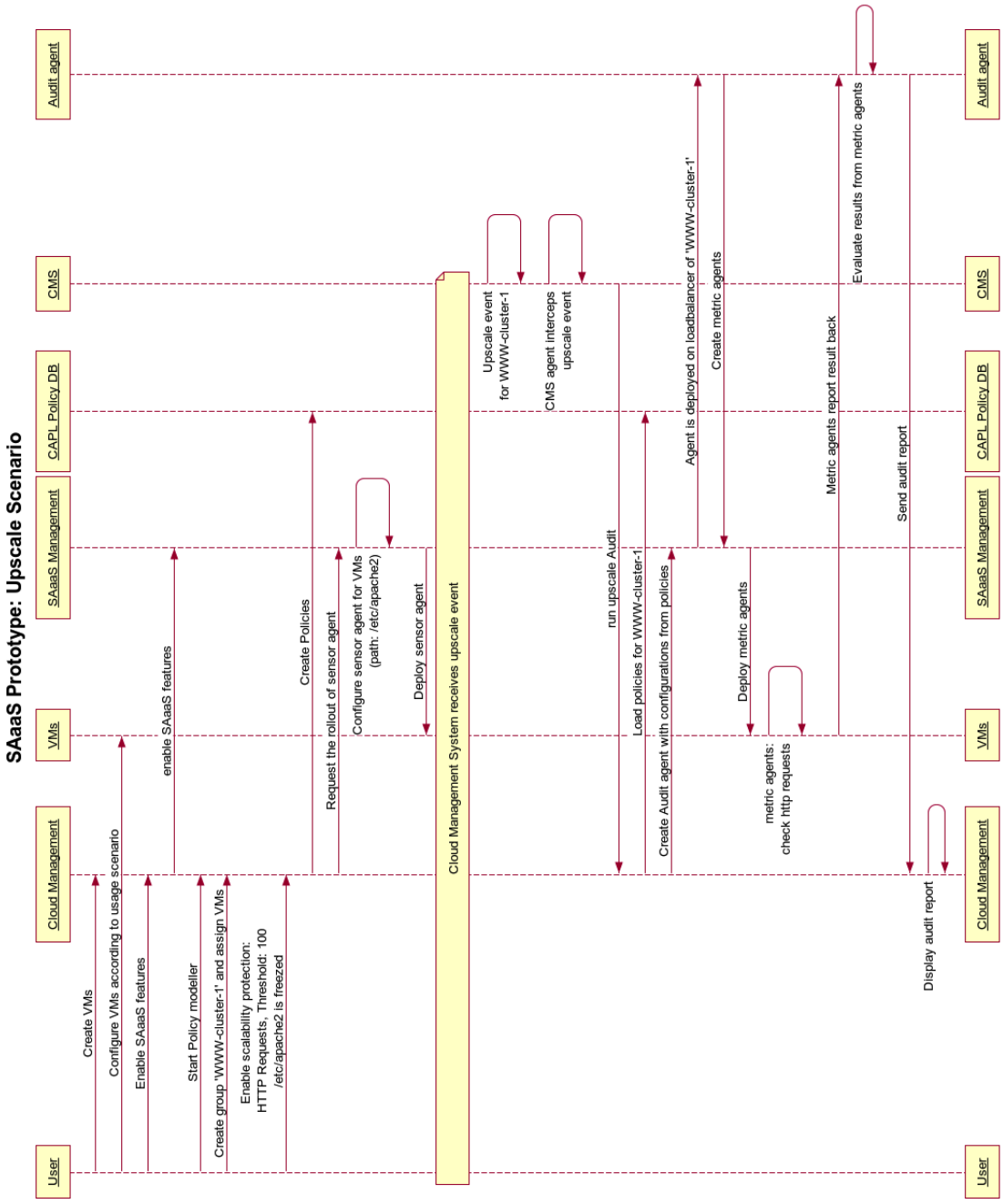


Figure A.2: SAaaS Demo2: Concurrent cloud audit in case of upscale event

A.4 Bootchart Analysis of SAaaS Agents Platform Overhead

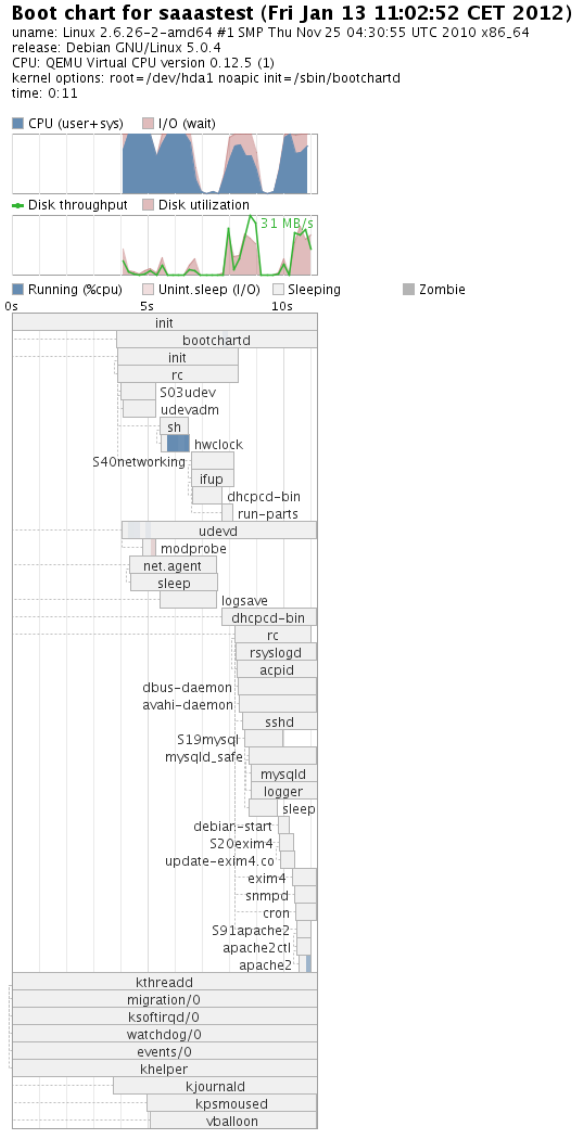


Figure A.3: Bootchart analysis of VM without SAaaS agent platform

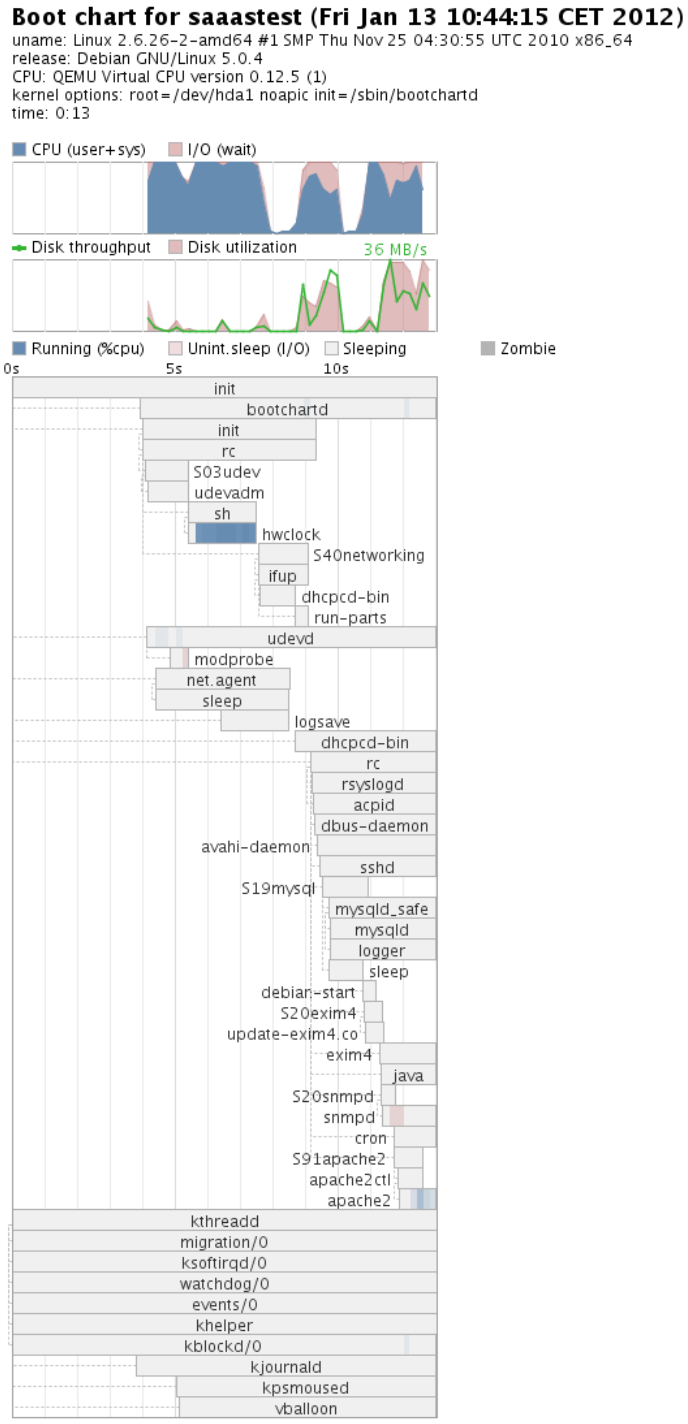


Figure A.4: Bootchart analysis of VM with SAaaS agent platform

A.5 Email Correspondence Public Cloud Providers

Email Correspondence with IBM

from: Frank Doelitzscher <doelitz@hs-furtwangen.de>
To: Elisabeth Kolbe/Germany/IBM@IBMDE
Date: 17.01.2012 16:16
Subject: Cloud Anfrage

Sehr geehrte Frau Kolbe,

ich suche einen Datensatz mit aufgezeichnete Nutzungsdaten aus einer Cloud Infrastruktur. Ich vermute den findet man im Team um Eure Cloud Referenz Architektur!

Hintergrund:
Wir am Cloud Research Lab arbeiten gerade an einer Anomalieerkennung von Cloud-Nutzungsverhalten und versuchen dafür verschiedene Algorithmen zu entwickeln. Um deren Funktion und Effizienz testen zu können, wäre ein Nutzungsdatensatz aus einer größeren Cloud Umgebung hilfreich.

Wie ich auf Sie gekommen bin:
Ich habe auf der IEEE CloudCom 2011 in Athen den Vortrag "Deconstructing Amazon EC2 Spot Instance Pricing" gehört. Da ging es auch um Auswertung von Cloud-Nutzungsdaten und die Referentin erzählte mir, dass sie in Ihrem aktuellen Projekt mit IBM zusammenarbeitet und hier einen guten Cloud-Nutzungsdatensatz als Grundlage für Ihre Forschung verwendet.

Wir sind hier natürlich immer auch generell an einer Zusammenarbeit mit der Industrie interessiert. Vorstellbar wären gemeinsame Publikationen oder Arbeiten auf dem Gebiet. Ggf. benötigte Non Disclosure Agreements, z.B. bei der Verwendung eines Nutzungsdatensatzes können wir gern unterzeichnen.

Viele Grüße,
Frank Dölitzscher
--

Frank Dölitzscher
Doktorand
Fakultät Informatik
Hochschule Furtwangen - Informatik, Technik, Wirtschaft, Medien
Robert-Gerwig-Platz 1,
D-78120 Furtwangen, Germany
Phone: +49 (0)7723 920-2326
Frank.Doelitzscher@hs-furtwangen.de
http://www.wolke.hs-furtwangen.de

----- Forwarded by Elisabeth Kolbe/Germany/IBM on 17.01.2012 22:45 -----

From: Elisabeth Kolbe" <elisabeth.kolbe@de.ibm.com>
Date: 17.01.2012 20:16
Subject: Re: Cloud Anfrage

Hallo Herr Rindle, Hallo Herr Behrendt, Hallo Herr Betzler,

anbei findet ihr die Anfrage eines befreundeten Doktoranden von der FH Furtwangen, der sich gerade mit dem Thema Cloud Security beschäftigt:
"Hintergrund:
Wir am Cloud Research Lab arbeiten gerade an einer Anomalieerkennung von Cloud-Nutzungsverhalten und versuchen dafür verschiedene Algorithmen zu entwickeln. Um deren Funktion und Effizienz testen zu können, wäre ein Nutzungsdatensatz aus einer größeren Cloud Umgebung hilfreich."

>> Könnt ihr ihm ihn unterstützen bzw. habt ihr entsprechende Ansprechpartner, die weiterhelfen können?
Die komplette Anfrage findet ihr unten in der Mail - es wäre auch eine weitere Zusammenarbeit denkbar, NDA's sind selbstverständlich.
Nähere Informationen zum Projekt sind auf folgender Webseite zu finden:
http://www.wolke.hs-furtwangen.de/currentprojects/clouddatasec

Mit freundlichen Grüßen / Kind regards

Elisabeth Kolbe

Consultant, IT Management Consulting
Global Technology Services

Phone: +49-170-7995702
E-Mail: elisabeth.kolbe@de.ibm.com

IBM Deutschland
IBM-Allee 1
71139 Ehningen
Germany

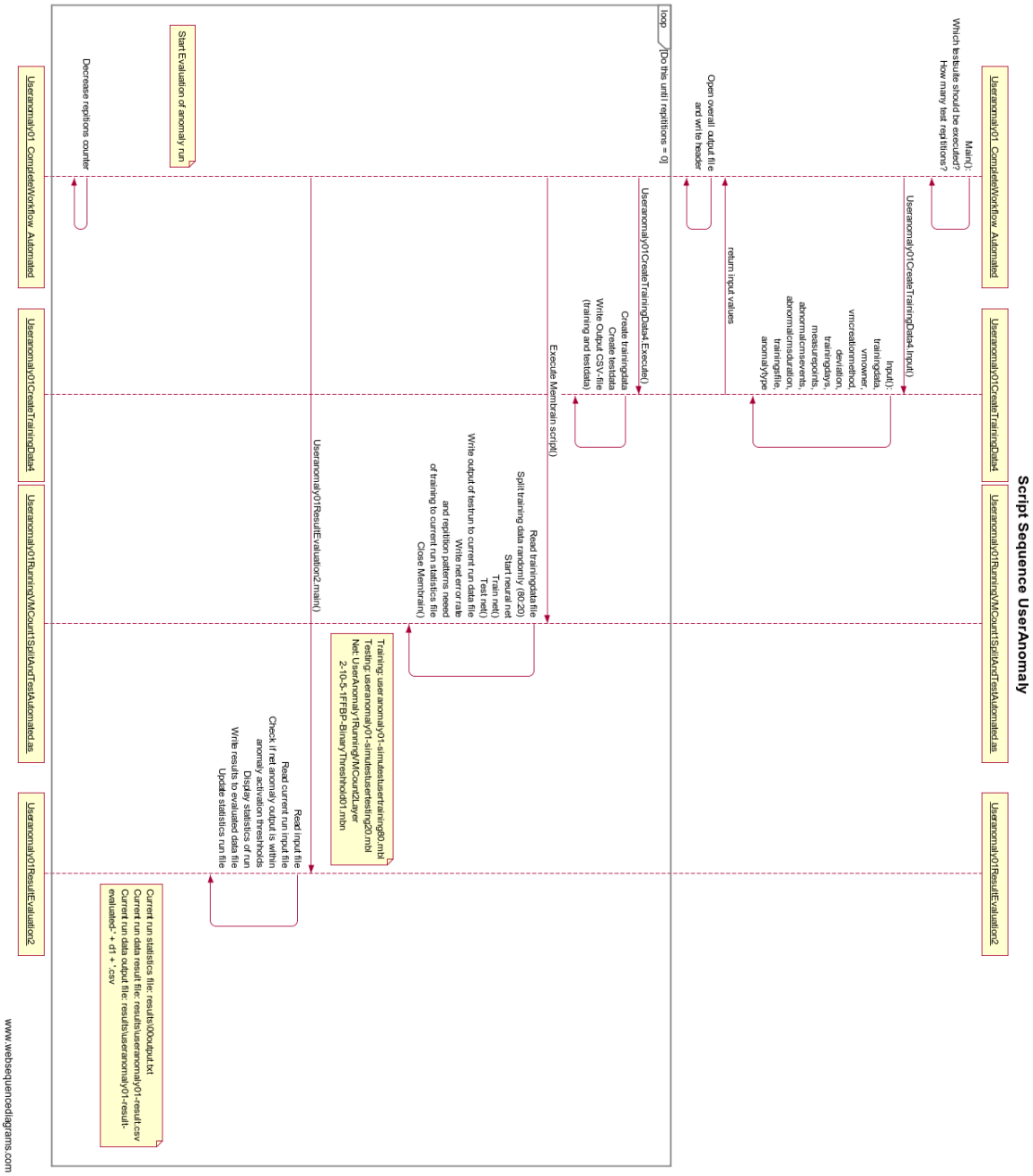
<0.DAE.gif>

From: Boas Betzler/Germany/IBM
To: Elisabeth Kolbe/Germany/IBM@IBMDE
Date: 18.01.2012 15:20
Subject: Re: Fw: Cloud Anfrage

Frau Kolbe, wir geben grundsätzlich keine Cloud Nutzungsdaten heraus.

Boas Betzler, IBM Senior Technical Staff
Member of the IBM Academy of Technology
IBM Global Services Master Inventor
cell: +49 1578 7508178

A.6 Simulation sequence scenario “UserAnomaly01”



A.7 Conference Poster

Validating Cloud Infrastructure Changes by Cloud Audits

Validating Cloud Infrastructure Changes By Cloud Audits

Frank Doelitzscher*, Christian Fischer*, Denis Moskal*, Christoph Reich*, Martin Knahl* and Nathan Clarke°

*Furtwangen University - Cloud Research Lab
Robert-Gerwig-Platz 1, 78120, Furtwangen,
Germany

°Centre for Security, Communications and Network
Research - University of Plymouth Plymouth PL4 8AA,
United Kingdom

Introduction

One characteristic of an Infrastructure as a Service (IaaS) cloud is its dynamic nature. New virtual machines (VMs) get deployed, existing VMs migrate to a different host or network segment and VMs vanish since they get deleted by their users. Classic incidence monitoring mechanisms are not flexible enough to cope with these frequent infrastructure changes. To adapt IT security audits to cloud computing the Cloud Research Lab at University of Applied Sciences Furtwangen is participating in a collaborative research project "***Security Audit as a Service***" funded by the German Ministry for Education and Research (BMBF) to investigate how audits of cloud infrastructure can enhance transparency and therefore trust in cloud environments. The developed "Security Audit as a Service (SAaaS)" prototype presented in this work and the appendant SERVICES CUP 2012 demonstration shows how autonomous agents can react on cloud infrastructure changes and automatically re-validate the security status and compliance of IT-security policies after the infrastructure change. This results in a ***concurrent cloud audit***.

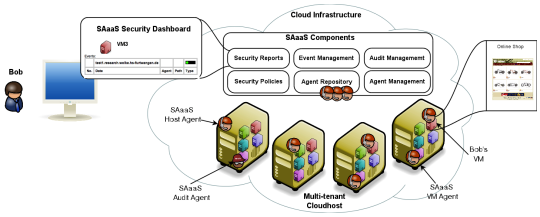


Figure 1: SAaaS components and initial situation of SAaaS demo

Security Audit as a Service Demo scenario

To show how concurrent cloud audits can be achieved, the following demonstration for the Services Cup 2012 has been developed: Bob, a web administrator of a company, runs a VM containing a web server hosting an online shop. At the beginning of the demo, the security status of Bob's VMs complies with Bob's IT-Security policies, which corresponds to an audit security status of green, shown in the SAaaS Dashboard. During the demo, Bob will deploy a new web server VM with a web server configuration which violates Bob's IT security policies (e.g. web server must be configured for SSL communication usage). The SAaaS architecture will detect this through concurrent security audits and sets the security status to red.

Demo Step by Step

Figure 2: Audit reports

Step1: Bob has already one VM deployed and an security audit was performed. Since the VM is compliant to Bob's IT security policies, the audit report shows no errors and a simple security indicator light shows green, as depicted in Figure 1 on the left.

Step 2: The user creates a new VM based on the SAaaS web server template. Automatically SAaaS default agents as well as web server specific sensor agents will be deployed to the VM and monitor its security state.

Step 3: The user alters the web server configuration in a way which violates a security policy. A running sensor Agent (inotify Agent) on the VM notices the changes and sends an event to the local Event Aggregator Agent (see Figure 4 for details on an an agent's structure).

Step 4: The Event Aggregator Agent informs a central SAaaS Manager Agent about this infrastructure change. It initiates the deployment of a new Audit Agent to verify the compliance of the new VM to Bob's security policies. A new Audit Agent gets configured and deployed to the new VM. The Audit Agent detects an invalid web server configuration and sends his report to the Manager Agent. These events are periodically queried by the SAaaS Security Dashboard and displayed.

Step 5: A new audit report gets displayed. Due to the violating web server configuration, the security indicator shows red in the security audit report, as depicted in Figure 2.

Acknowledgements
This research is supported by the German Federal Ministry of Education and Research (BMBF)

Automatic Security Audit after detected cloud infrastructure change

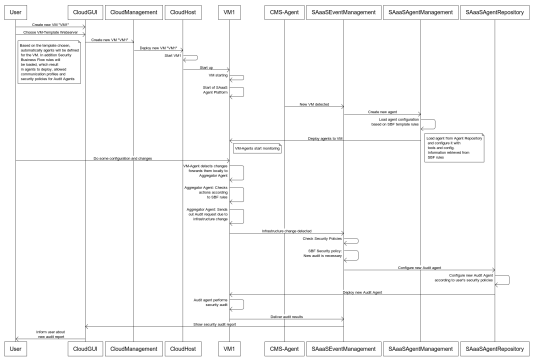


Figure 3: Information Sequence in case of an cloud infrastructure change

Figure 3 describes the information flow in case of a cloud infrastructure change. A user deploys a new VM, choosing the web server VM template. This infrastructure change gets detected by a SAaaS agent monitoring the Cloud Management System (CMS). Therefore, corresponding agents according to the template are deployed to the VM. Then, the user alters the VMs configuration. This gets detected by a sensor agent, and reported. Security policies demand in this case for a re-validation of the security status. A new Audit Agent gets configured and automatically deployed to the VM. Results are conditioned into a security audit report, shows at the SAaaS dashboard.

Security Audit as a Service - Agent Structure

Agents live in an agent platform, which provides them with basic services such as message delivery. A platform is composed of one or more Containers. Containers can be executed on different hosts, thus achieving a distributed platform. Each container can contain zero or more agents. To provide monitoring functionality a VM agent interacts through agent plugins with standalone tools, like a process monitor, intrusion detection system or anti virus scanner, as depicted in Figure 4. To harness the potential of cloud computing an agent can be deployed to a VM on-demand according to theSecurity Business Flow rules a user defines. Different agents based on modeled business processes are stored within an agent repository. To be able to move an agent to a running cloud instance an Inter Platform Mobility Service was integrated. This supports the advantage of deploying agents on-demand if a designed business process flow was started

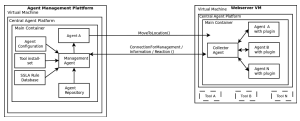


Figure 4: SAaaS agent architecture

Security Audit as a Service - Evaluation

Cloud Security Issue	Mitigation by SAaaS	SAaaS component
Abuse of cloud resources	Detection of compromised VMs Customer independent attacks	VM agents Behaviour analysis
Attacks on cloud scalability	Detection of malicious scalability commands Control of SLA compliance Control of cloud usage	SBF rules SBF rules, agents SBF rules
Missing security monitoring (customer view)	Overview over cloud resources Continuous security status of cloud instances Monitoring of provider compliance Better risk evaluation in case of incidents	Agents Concurrent audits Host agents
Missing security monitoring (provider view)	Detection of knock-on effects Detection of attacks on CMS Detection of customer overspanning attacks Interface for (external) cloud audit	Agents Agents, SBF rules Behaviour analysis

Publication & Information
You can find more information about our work at: http://www.wolke.hs-furtwangen.de
The full version of the paper can be found in the proceedings of the 2012 IEEE Congress on Services (SERVICES 2012) which is collected in IEEE Digital Library.